

NASA TECHNICAL NOTE



NASA TN D-3742

0.1

LOAN COPY: RETURN
AFWL (WLIL-2)
KIRTLAND AFB, NM



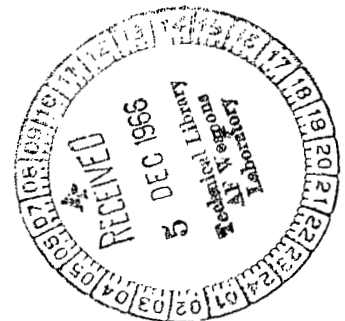
NASA TN D-3742

EXPERIMENTAL PERFORMANCE EVALUATION OF A RADIAL-INFLOW TURBINE OVER A RANGE OF SPECIFIC SPEEDS

by Milton G. Kofskey and Charles A. Wasserbauer

Lewis Research Center

Cleveland, Ohio





0130499

NASA TN D-3742

EXPERIMENTAL PERFORMANCE EVALUATION OF A RADIAL-INFLOW
TURBINE OVER A RANGE OF SPECIFIC SPEEDS

By Milton G. Kofskey and Charles A. Wasserbauer

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 - Price \$1.00

EXPERIMENTAL PERFORMANCE EVALUATION OF A RADIAL-INFLOW TURBINE OVER A RANGE OF SPECIFIC SPEEDS

by Milton G. Kofskey and Charles A. Wasserbauer

Lewis Research Center

SUMMARY

An experimental investigation was made to determine the effect of specific speed on efficiency for a 4.59-inch radial-inflow turbine. The range of specific speeds investigated (72 to 108) at equivalent design speed and pressure ratio was obtained by changing volume flow, based on rotor exit conditions. Changes in volume flow were accomplished by the use of stators having throat areas nominally 50, 75, 100, and 125 percent of design. The turbine was operated with air as the working fluid.

Maximum total and static efficiencies were obtained over the specific speed range of about 80 to 90. The peak total and static efficiencies were 0.91 and 0.87, respectively, for the 75-percent configuration.

An understanding of the losses which contributed to the variation of turbine performance with specific speed at design blade-jet speed ratio was made possible by an analysis which determined the magnitude of the various losses for each configuration. Stator loss was the predominant contributor to the decrease in efficiency as specific speed was reduced from a value of 86. Rotor incidence and viscous losses were the primary contributors to the decrease in performance when specific speed was increased above the value of 90. Stator exit static-pressure measurements showed that, at equivalent design speed and pressure ratio, rotor reaction increased as specific speed increased.

Rotor exit total-pressure and flow angle surveys indicated that low losses were obtained near the hub region of the rotor for all configurations at equivalent design speed and pressure ratio. Comparatively high losses were obtained near the outer wall. These increased losses may have resulted from tip leakage effects with blade unloading, as well as from centrifugation of low-momentum fluid to this region.

INTRODUCTION

The current Brayton-cycle space-power technology program at the Lewis Research Center includes the experimental investigation of factors which influence the performance of small radial-inflow turbines. One such factor is the specific speed parameter, which relates the operating variables of turbine rotative speed, volume flow based on exit conditions, and ideal specific work to turbine geometry and aerodynamic performance.

Reference 1 shows specific speed - efficiency correlations for a number of radial-inflow turbines of various sizes and for a wide range of inlet conditions. This reference shows that high efficiency is attainable for a specific speed range from 65 to 105, with a significant reduction in efficiency outside this range. However, turbine size, rotor tip clearance, and Reynolds number effects are present in the specific speed - efficiency correlations but are not examined separately. Therefore, the experimental investigation described herein was conducted to determine the specific speed effect on performance for a particular turbine size with rotor tip clearance and Reynolds number held constant.

Two approaches were considered to achieve the range of specific speeds. One was to design and fabricate an optimized stator and rotor configuration for each specific speed point, and the other was to use several stators with one rotor configuration. The second approach was chosen because it would minimize time and cost of the program; however, less than optimum turbine configurations may have resulted, especially at the extremes of the specific speed range.

The 4.59-inch-tip-diameter radial-inflow turbine of reference 2 was chosen as the research turbine. The design specific speed for this turbine is 95.6. Three additional stators having throat areas nominally 50, 75, and 125 percent of design were fabricated. The four configurations cover a specific speed range of 68 to 107 at equivalent design speed and pressure ratio. Each configuration was investigated over a range of turbine pressure ratios at equivalent design speed.

This report presents the performance of the subject turbine for each configuration and shows the specific speed effect on turbine efficiency. Results are presented in terms of equivalent weight flow and efficiency at equivalent design speed over a range of pressure ratios. Internal flow characteristics are presented in terms of static pressure variation through the turbine and radial variation of exit flow angle and loss distribution at the rotor exit.

SYMBOLS

g gravitational constant, 32.174 ft/sec^2

H' isentropic specific work based on total-pressure ratio, ft-lb/lb

Δh	specific work, Btu/lb
J	mechanical equivalent of heat, 778.029 ft-lb/Btu
N	turbine speed, rpm
N_s	specific speed, $NQ^{1/2}/(H^3)^{3/4}$, ft ^{3/4} /(min)(sec ^{1/2})
p	pressure, psia
Q	volume flow (based on exit conditions), cu ft/sec
Re	Reynolds number, $w/\mu r_t$
r	radius, ft
U	blade velocity, ft/sec
V	absolute gas velocity, ft/sec
V_j	ideal jet-speed corresponding to total- to static-pressure ratio across turbine, $(2gJ \Delta h_{id})^{1/2}$, ft/sec
W	relative gas velocity, ft/sec
w	weight flow, lb/sec
α	absolute rotor exit gas flow angle measured from axial direction, deg
γ	ratio of specific heats
δ	ratio of inlet total pressure to U. S. standard sea-level pressure, p_1/p^*
ϵ	function of γ used in relating parameters to that using air inlet conditions at U. S. standard sea-level conditions, $\frac{0.740}{\gamma} \left(\frac{\gamma + 1}{2} \right)^{\gamma/\gamma-1}$
η_s	static efficiency (based on total- to static-pressure ratio across turbine)
η_{tot}	total efficiency (based on total- to total-pressure ratio across turbine)
θ_{cr}	squared ratio of critical velocity at turbine inlet to critical velocity at U. S. standard sea-level temperature, $(V_{cr,1}/V_{cr}^*)^2$
μ	gas viscosity, lb/(ft)(sec)
ν	blade-jet speed ratio (based on rotor inlet tip speed), U_t/V_j

Subscripts:

cr	condition corresponding to Mach number of unity
id	ideal
w	outer wall

- t tip
- 1 station at turbine inlet
- 2 station at stator exit
- 3 station at turbine exit

Superscripts:

- ' absolute total state
- * U. S. standard sea-level conditions (temperature equal to 518.67° R and pressure equal to 14.696 psia)

TURBINE DESCRIPTION

The 4.59-inch-tip-diameter radial-inflow turbine described in reference 2 was selected for this investigation. Air equivalent design values are as follows:

Equivalent weight flow, $w\epsilon\sqrt{\theta_{cr}}/\delta$, lb/sec	0.616
Equivalent specific work, $\Delta h/\theta_{cr}$, Btu/lb	11.9
Equivalent speed, $N/\sqrt{\theta_{cr}}$, rpm	29 550
Equivalent total- to static-pressure ratio, p'_1/p_3	1.540
Total to total efficiency, η_{tot}	0.880
Total to static efficiency, η_s	0.824
Blade-jet speed ratio, ν	0.697
Specific speed, N_s , $NQ^{1/2}/(H')^{3/4}$, $\text{ft}^{3/4}/(\text{min})(\text{sec}^{1/2})$	95.6

The range of specific speeds at equivalent design speed and pressure ratio was obtained by changing volume flow by using stators with different throat areas. This was done by essentially changing the stator blade angle. Three additional stators having nominal throat areas of 50, 75, and 125 percent of design were used to obtain nominal specific speeds of 68, 83, and 107.

Figure 1 shows the four stators and the rotor used in the investigation. The measured stator throat areas were 49.6, 75.3, 96.1, and 126.1 percent of design. Hereafter, each stator and rotor combination will be referred to as the 50-, 75-, 100-, and 125-percent configuration. One stator blade of each configuration had an elongated leading edge to block the flow from entering the small area end of the inlet scroll. A description of the 100-percent configuration including velocity diagrams is given in reference 2. The 100- and 125-percent stators each have 14 blades, whereas the 50- and 75-percent stators have 18 blades each. In order to maintain acceptable stator-blade

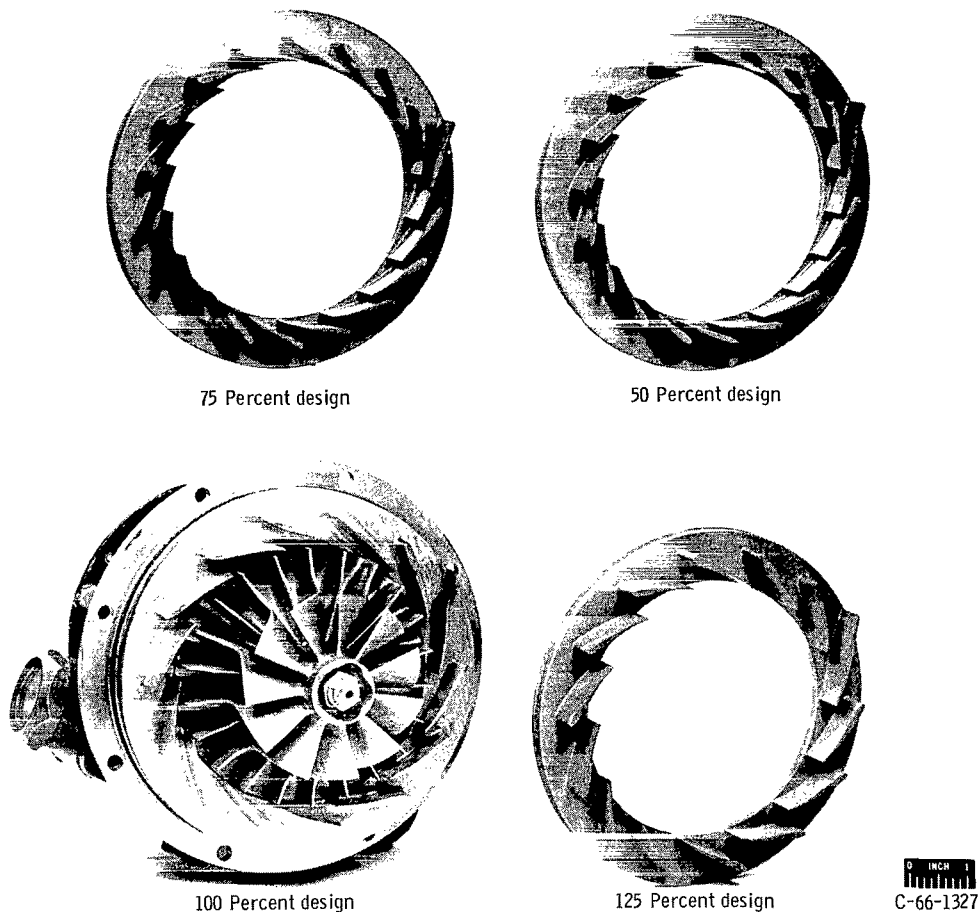


Figure 1. - Turbine rotor and four stators.

surface velocities, the 50-, 75-, and 125-percent stators have slightly different shapes than the 100-percent stator.

It may be noted that, although the throat area of the 100-percent stator was 3.9 percent smaller than design, results (as reported in ref. 2) showed that equivalent design weight flow was obtained at equivalent design speed and pressure ratio. Attainment of equivalent design weight flow results from the flow check procedure, in which the rotor throat area is increased by cutting back rotor trailing edges until equivalent design weight flow is obtained.

The rotor has 11 blades and 11 splitter vanes. These splitter vanes are used over the initial third of the rotor, thereby increasing the solidity in this region. The resultant decrease in loading was required at the hub to prevent low blade pressure-surface gas velocities.

APPARATUS, INSTRUMENTATION, AND METHODS

The test facility, instrumentation, and method of calculating performance parameters were the same as those described in reference 2, except that air was used as the working fluid. Figure 2 shows a cross-sectional sketch of the turbine test section and the instrument measuring stations. A varying area scroll was used to obtain uniform inlet conditions at the stator inlet. A center body was used at the rotor exit to obtain measurement of exit static pressure at the hub and at the outer wall. Radial surveys of total pressure, total temperature, and flow angle were made at the rotor exit.

The 100-percent configuration was tested at nominal inlet conditions of 16.0 pounds per square inch absolute and 540° R and resulted in a weight flow of 0.657 pound per second at equivalent design speed and pressure ratio. A nominal Reynolds number of 277 000 was calculated from this result; Reynolds number is defined herein as $Re = w/\mu r_t$. In order to eliminate the effects of changes in Reynolds number on turbine efficiency, this parameter was held constant for all configurations at equivalent design speed and pressure ratio. Thus, the inlet total pressure was adjusted for the other configurations until a weight flow of approximately 0.657 pound per second was obtained. Table I shows the values of inlet total pressure and temperature and the pressure ratio

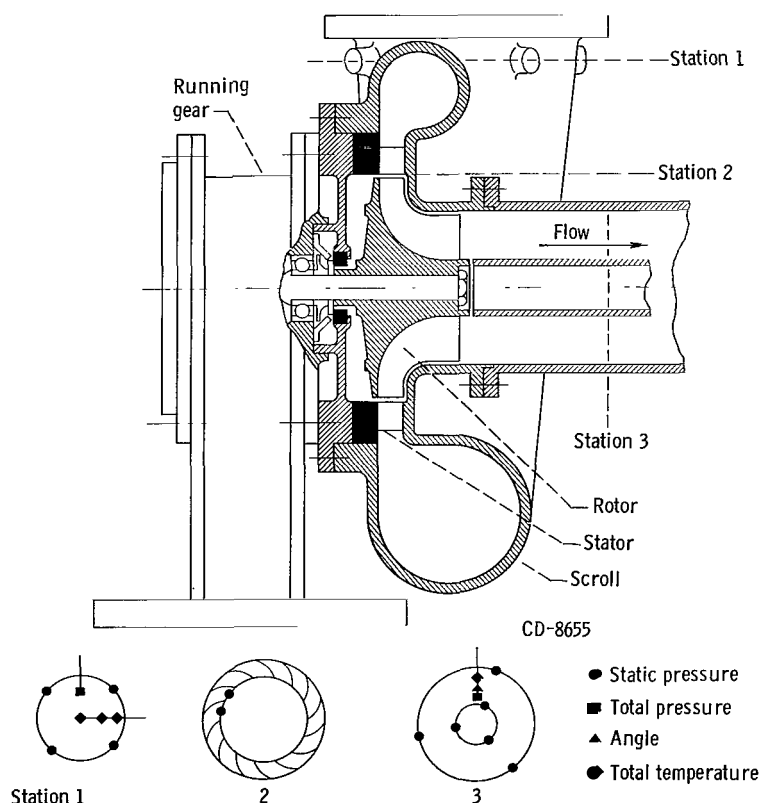


Figure 2. - Turbine test section and instrumentation.

TABLE I. - EXPERIMENTAL OPERATING CONDITIONS

Configuration, percent design	Inlet total pressure, psia	Inlet total temperature, °R	Pressure-ratio range
125	13.0	536	1.28 to 2.13
100	16.0	540	1.30 to 2.16
75	19.2	540	1.29 to 2.26
50	27.2	542	1.31 to 2.32

range over which the turbine was investigated for each configuration.

The turbine was rated on the basis of both total and static efficiency.

Turbine inlet and exit total pressures were calculated from weight flow, static pressure, total temperature, and flow angle. In the calculations of turbine inlet total pressure, the flow was

assumed to be normal to the plane defined by station 1. The exit total temperature was determined from turbine power measurements.

RESULTS AND DISCUSSION

The results of this investigation are presented in two sections. The first section includes overall results in terms of equivalent weight flow and efficiency for a range of pressure ratios at equivalent design speed with cold air as the working fluid. The effect of specific speed on turbine efficiency is then shown. The second section discusses the internal flow characteristics of the turbine as determined from exit radial surveys of angle and total- and static-pressure measurements through the turbine at equivalent design speed and pressure ratio.

Turbine Performance

Weight flow. - Figure 3 shows the variation of equivalent weight flow $w \epsilon \sqrt{\theta_{cr}}/\delta$ with inlet total- to exit static-pressure ratio at equivalent design speed. Equivalent weight flows of 0.752, 0.615, 0.519, and 0.367 pound per second were obtained for the 125-, 100-, 75-, and 50-percent configurations at the equivalent design pressure ratio of 1.54. The variation of weight flow with increasing pressure ratio indicated that the flow was subsonic over the entire range of pressure ratios covered. The figure also shows that near choked flow conditions were obtained for the 50-percent configuration at the pressure ratio of 2.32. The combination of near choked flow conditions obtained for the 50-percent configuration and the flattening of the weight-flow curves with decreasing stator-throat area indicates that the velocity level through the stator blade row was increasing with decreasing stator throat area.

Figure 4 presents the variation of equivalent weight flow with stator throat area for equivalent design speed and pressure ratio. Equivalent weight flow is expressed as

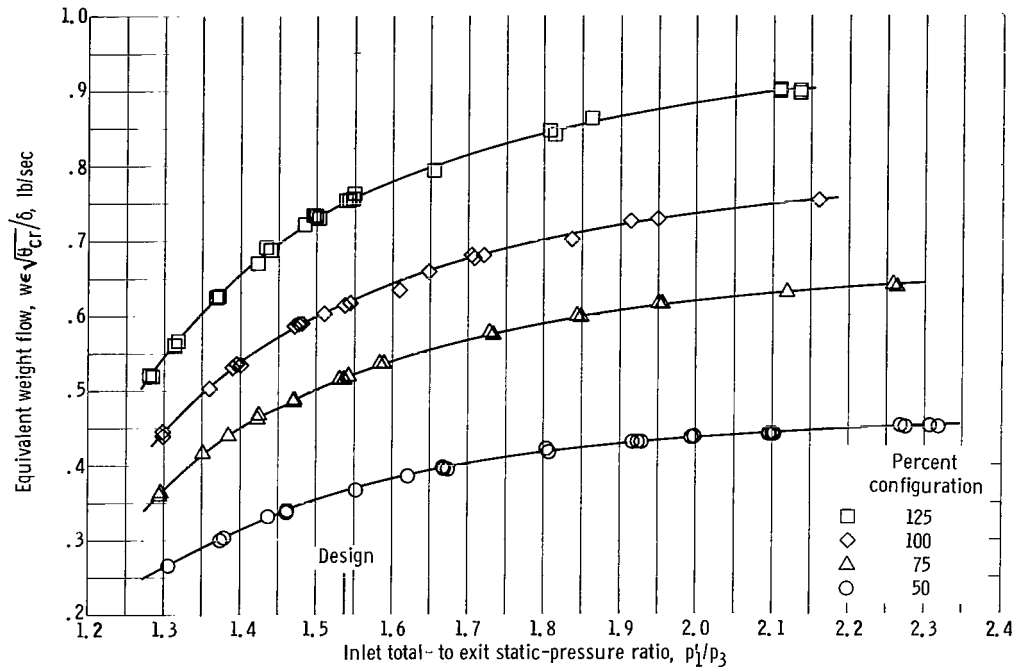


Figure 3. - Variation of weight flow with pressure ratio and stator throat area at equivalent design speed.

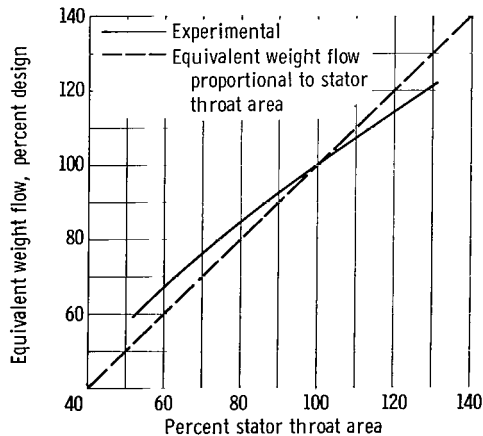


Figure 4. - Variation of weight flow with stator throat area at equivalent design speed and pressure ratio. (Based on measured throat area of 100-percent stator.)

a percentage of the experimental equivalent weight flow obtained with the 100-percent configuration. The dashed line shown on the figure represents the case where equivalent weight flow is directly proportional to stator throat area. Comparison of the experimental curve with the ideal case shows that the weight flow increases at a lower rate than the rate of area increase. This indicates that the stator pressure ratio p_2/p_1 increased with increasing stator throat area and, therefore, rotor reaction increased. This change in rotor reaction resulted from the variation of stator to rotor throat area ratios of the four configurations. The change in rotor reaction among the four configurations is discussed further in the section Internal Flow Characteristics.

Efficiency. - Figure 5 shows the variation of total and static efficiency with blade-jet speed ratio for each configuration. The highest efficiencies, at design blade-jet speed ratio, were obtained with the 75-percent configuration. Total and static

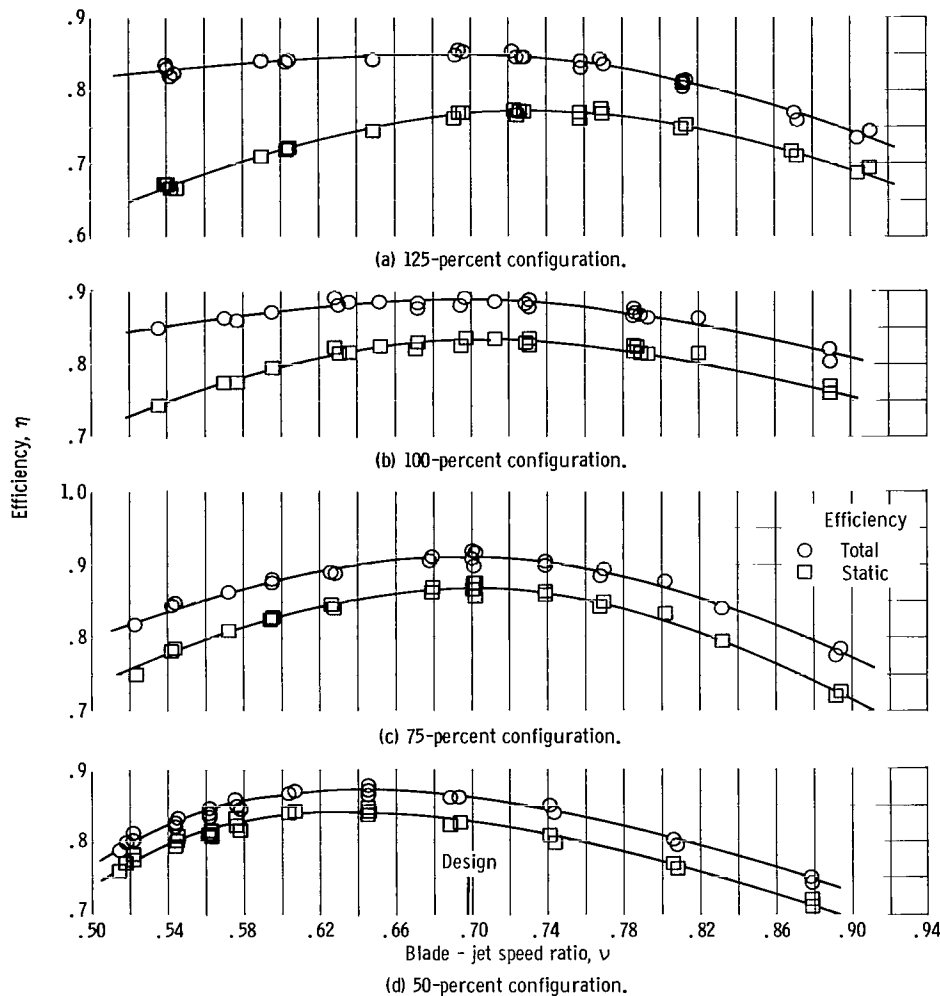


Figure 5. - Variation of efficiency with blade - jet speed ratio at equivalent design speed.

efficiencies were 0.91 and 0.87, respectively, for this configuration. These values are significantly higher than the total and static efficiencies of 0.89 and 0.83 obtained with the 100-percent configuration. At design blade-jet speed ratio, the lowest efficiencies, total and static, were obtained with the 125-percent configuration. These values were 0.85 and 0.77 for the total and static efficiencies, respectively.

The level of rotor exit velocity, as indicated by the difference between total and static efficiency, decreases with decreasing stator throat area. For example, at the design blade-jet speed ratio of 0.697, approximately 8 points in efficiency are attributed to rotor kinetic energy for the 125-percent configuration, while only 3.0 points in efficiency are attributed to rotor exit kinetic energy for the 50-percent configuration. This decrease in rotor exit velocity with decreasing stator throat area results from the change in the stator to rotor throat area ratio among the four configurations. The figure also indicates the variation of rotor exit kinetic energy with blade-jet speed ratio. Comparison

of figures 5(a) and (d) shows that there was a greater rate of change in exit kinetic energy with increasing blade-jet speed ratio (decreasing turbine pressure ratio p'_1/p_3) for the 125-percent configuration than for the 50-percent configuration. This effect results from the larger variation of weight flow with pressure ratio for the 125-percent configuration than for the other configurations, as shown in figure 3 (p. 8).

Figure 6 shows the variation of total and static efficiency with specific speed for all four configurations investigated. The dashed line represents the variation of efficiency with specific speed at the design blade-jet speed ratio of 0.697. The upper plot in figure 6 shows that the highest total efficiency value of 0.91 was obtained at a specific speed of approximately 86. This efficiency value is 2.0 points higher than the efficiency of 0.89, which was obtained at the design specific speed value of 95.6 for the 100-percent or reference turbine configuration. It may be noted that the design blade-jet speed ratio curve (dashed line) passes through the peak efficiency point for all but the 50-percent configuration. The heavy curve shown in the figure represents the envelope of the efficiency curves for all configurations. This curve shows that maximum total efficiency is obtained in the specific speed range of about 80 to 90.

The lower plot in figure 6 shows the variation of static efficiency with specific speed for the four configurations. The highest efficiency value of 0.87 was also obtained at a specific speed of approximately 86. This value of efficiency is about 3.0 points higher than that obtained for the 100-percent or reference turbine configuration at the design specific speed of 95.6. The lowest peak static efficiency of 0.77 was obtained at a

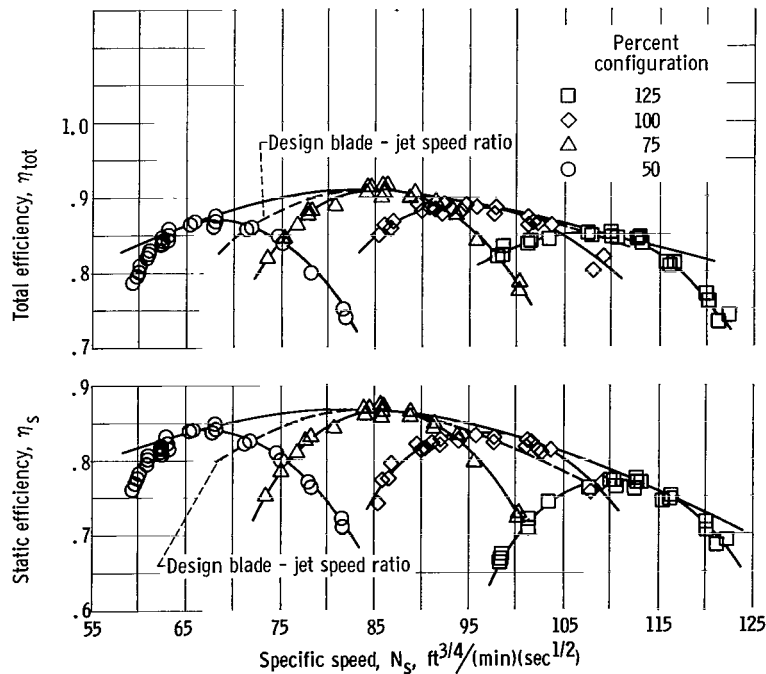


Figure 6. - Variation of efficiency with specific speed at equivalent design speed.

specific speed of 111. It should be pointed out, however, that part of this decrease in static efficiency results from using the same rotor with each stator. Since the 125-percent configuration passes the largest volume flow of the four configurations, the rotor exit kinetic energy would be expected to be higher for this configuration.

The variation of static efficiency with specific speed for design blade-jet speed ratio (dashed line) shows the same trend as the envelope curve represented by the heavy line. The highest efficiency of 0.87 was obtained at a specific speed of 86, and the lowest efficiency of 0.77 was obtained at a specific speed of 108. It may be noted that both total and static efficiencies obtained at design blade-jet speed ratio occur at or very close to the peak efficiency points for the 75- and 100-percent configurations and at lower values of efficiency for the other configurations. From these results, it appears that radial-inflow turbines should be designed for a specific speed range of about 80 to 90 for the attainment of high efficiency.

Loss distribution. - In order to obtain an understanding of the losses which contributed to the variation of turbine performance with specific speed at design blade-jet speed ratio, an analysis was made to determine the magnitude of the various losses for each configuration. The method used involved the determination of velocity diagrams for each configuration from measured turbine work, weight flow, inlet conditions of pressure and temperature, speed, stator throat area, and results of rotor exit surveys of total pressure and flow angle. Design loss distribution between the stator and rotor was used to proportion the measured overall turbine loss for the 100-percent configuration. Stator losses for the other configurations were then assumed to vary in proportion to the average of inlet and outlet kinetic energy as determined from the velocity diagrams.

Rotor incidence losses were determined through adjustment of the actual incidence angle, which resulted in an effective relative whirl velocity different from the velocity diagram value. The adjustment depends upon the blade speed, the number of blades, the rotor diameter, and the volume flow at the rotor inlet. The use of the effective relative whirl velocity is analogous to the use of the slip factor for centrifugal impellers. The remaining losses were attributed to rotor viscous losses. Figure 7 shows the results of these calculations. The various losses, expressed in terms of efficiency, are shown as a function of specific speed. The magnitude of the exit kinetic-energy loss is shown by the difference between total and static efficiency values obtained from figure 5 (p. 9) at design blade-jet speed ratio.

Figure 7 shows that rotor incidence loss increases as specific speed increases above 90. This increase in rotor incidence loss results from an increase in the stator exit flow angle (as measured from tangential) and a decrease in the stator exit velocity with increasing specific speed. Rotor viscous losses also increase substantially with increasing specific speed. The increase in rotor loss results from the increased relative velocity level through the rotor. Part of the increase in rotor viscous loss can be attributed to the manner in which the range of specific speeds was obtained. Figure 7

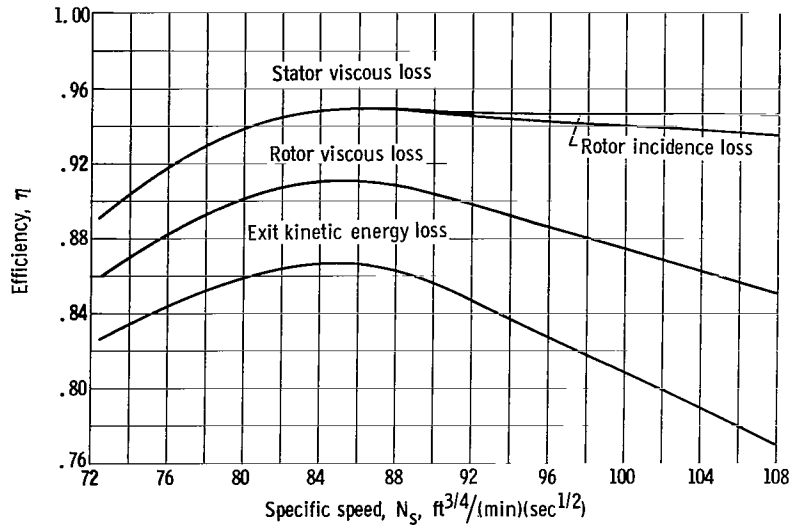


Figure 7. - Variation of turbine losses with specific speed at equivalent design speed and pressure ratio.

shows that there is no significant change in stator viscous losses as the specific speed is increased above a value of 86. This would indicate that the combined losses resulting from the velocity level through the stators and the boundary-layer blockage did not change to any large degree.

Decreasing specific speed below a value of 86 results in an increase in stator viscous losses. The figure shows that the losses increase from about 5.0 points in terms of efficiency at a specific speed of 86 to about 11.0 points at a specific speed of approximately 72. These losses may be associated with the increased velocity level through the stator and the increased boundary-layer blockage due to a larger ratio of wetted area to flow area.

Calculations also indicated that rotor incidence losses were insignificant below a specific speed of 90. Rotor viscous and exit kinetic-energy losses decreased with decreasing specific speed, since specific speed is proportional to the square root of the exit velocity when rotative speed, rotor throat area, and pressure ratio are constant.

Internal Flow Characteristics

The determination of turbine internal-flow characteristics for each configuration was based on the measured static-pressure distribution through the turbine, together with the results of a radial survey of turbine exit total pressure and flow angle.

Figure 8 shows the variation in stator exit static pressure with design stator throat area at equivalent design speed and pressure ratio for the four configurations investigated. Rotor reaction decreases and stator exit velocity increases with decreasing stator throat area, as was noted in the discussion of equivalent weight flow.

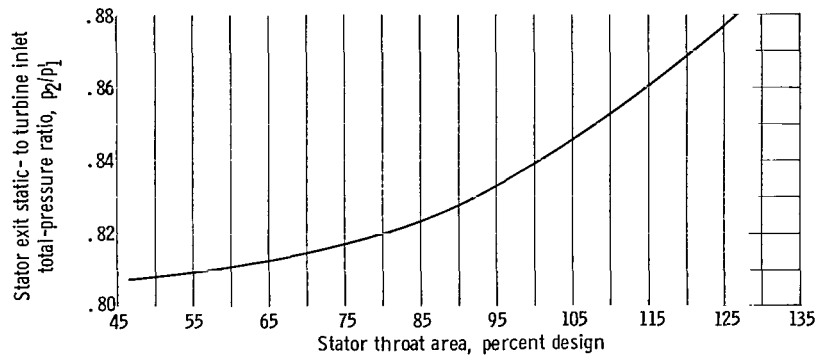


Figure 8. - Variation of stator exit static pressure with stator throat area at equivalent design speed and pressure ratio.

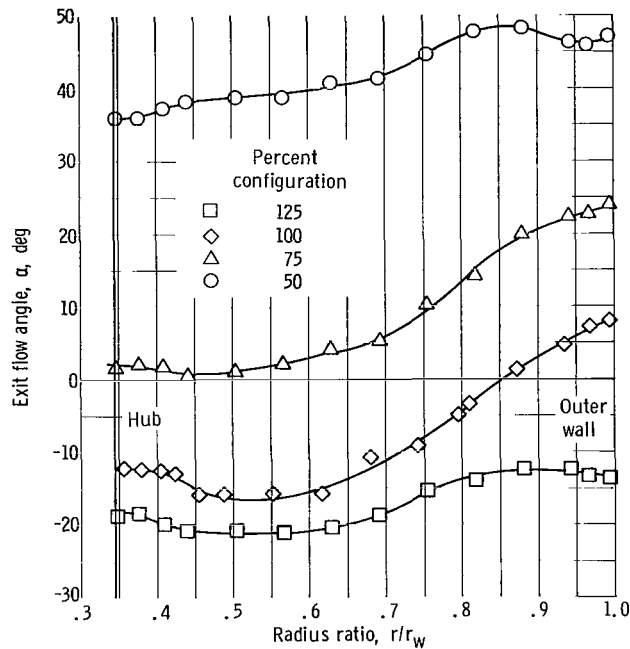


Figure 9. - Variation of turbine exit flow angle with radius ratio at equivalent design speed and pressure ratio.

The results of a radial survey of exit flow angle taken at equivalent design speed and pressure ratio are presented in figure 9 for the four configurations investigated. It may be noted that, as the stator throat area was reduced, the exit flow angle changed from predominately overturning (as denoted by negative angles) to underturning over the entire passage height. This trend in exit flow angle with configuration is to be expected since the rotor exit relative velocity decreases with decreasing stator throat area.

The variation of exit flow angle and exit total and static pressure with radius ratio indicated that there was a nonuniform work distribution from hub to outer wall for all four configurations, with minimum work occurring along the outer wall. This may be due to blade unloading which results from tip leakage and from centrifugation of low

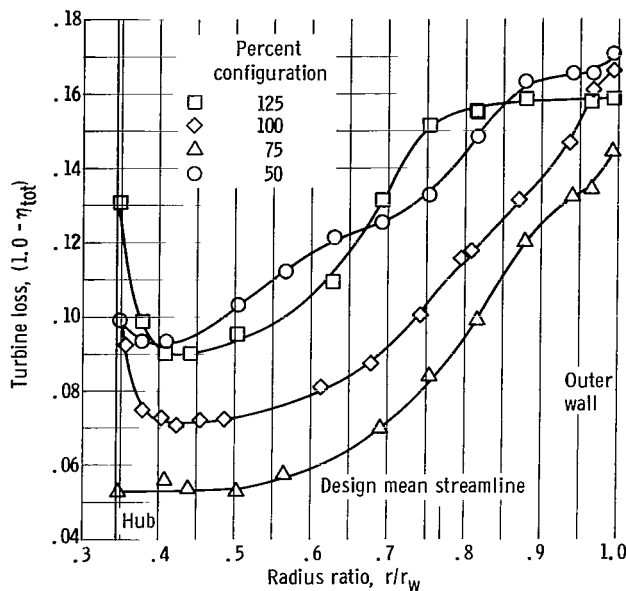


Figure 10. - Variation of turbine loss with radius ratio at equivalent design speed and pressure ratio.

momentum fluid to this region.

Local values of total efficiency were calculated on the basis of the change in tangential momentum through the rotor and the radial distribution of total pressure at the rotor exit. These results are plotted in figure 10 in terms of turbine loss $(1.0 - \eta_{tot})$ as a function of radius ratio. The figure shows that the largest radial variation in loss or efficiency is obtained with the 75- and 100-percent configurations. However, the magnitudes of the losses for these two configurations are substantially lower than those for the other two configurations. The curves show low losses along the hub region and comparatively high losses in the region near the outer wall for all configurations.

Calculations were made from experimental results to determine the radius ratio at which the weight flow was divided into equal parts. A radius ratio of approximately 0.77 was calculated for all configurations. This coincides with the design mean stream line, as shown in figure 10. At the mean stream line the calculated local loss is approximately equal to the experimental value as obtained from overall performance for each configuration.

SUMMARY OF RESULTS

An experimental investigation was made to determine the specific speed effect on performance for a 4.59-inch-tip-diameter radial-inflow turbine at equivalent design speed over a range of pressure ratios. Results are presented for operation at a nominally constant Reynolds number of 277 000 at equivalent design speed and pressure ratio. The effect of turbine size on performance was eliminated by use of the same rotor for each configuration. The range of specific speed values investigated at equivalent design speed and pressure ratio was obtained by changing volume flow through the use of stators having throat areas nominally 50, 75, 100, and 125 percent of design. From this investigation the following results were obtained:

1. Comparison of actual equivalent weight flow with an equivalent weight flow, which is directly proportional to stator throat area, showed that there was a deficiency in

weight flow for the 125-percent configuration and a surplus of weight flow for the 50- and 75-percent configurations at equivalent design speed and pressure ratio. This difference in weight flows was attributed to the corresponding changes in rotor reaction which result from the use of the same rotor for each configuration.

2. Maximum total and static efficiencies were obtained in the specific speed range of about 80 to 90. In this range, peak total and static efficiencies of 0.91 and 0.87 were obtained with the 75-percent configuration at a specific speed of 86. The lowest peak value of efficiency was obtained with the 125-percent configuration. For this case, the total efficiency was 0.85 at a specific speed of approximately 108. The corresponding static efficiency was 0.77 at a specific speed of 111.

3. An analysis of stator and rotor losses over the range of specific speeds investigated at equivalent design speed and pressure ratio showed the following:

- (a) Turbine losses were at a minimum for the specific speed range of about 80 to 90.
- (b) Stator viscous losses were the predominant factor in the decrease in total efficiency as the specific speed was decreased from a value of 86.
- (c) Rotor incidence and viscous losses were the predominant factors in the decrease in total efficiency as specific speed was increased from 90.

4. Stator-exit static-pressure measurements obtained at equivalent-design speed and pressure ratio indicated that the highest rotor reaction was obtained for the 125-percent configuration. Rotor reaction decreased with decreasing stator throat area.

5. Radial surveys of rotor exit total pressure and flow angle at equivalent design speed and pressure ratio indicated that minimum losses occurred near the hub region and the losses increased substantially near the outer wall. These increased losses may have resulted from tip leakage effects with blade unloading and from centrifugation of low-momentum fluid toward the outer wall. Comparison of losses between configurations showed that minimum losses were obtained from hub to outer wall for the 75-percent configuration.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, August 26, 1966,
120-27-03-13-22.

REFERENCES

1. Wood, Homer J. : Current Technology of Radial-Inflow Turbines For Compressible Fluids. J. Eng. Power, vol. 85, no. 1, Jan. 1963, pp. 72-83.
2. Wasserbauer, Charles A. ; Kofskey, Milton G. ; and Nusbaum, William J. : Cold Performance Evaluation of a 4.59-Inch Radial-Inflow Turbine Designed for a Brayton-Cycle Space Power System. NASA TN D-3260, 1966.

m_i	mass of i^{th} perturbing body, sun mass units
m_r	mass of reference body plus m , sun mass units
N_M	Mach number
P	atmospheric pressure, newtons/m ²
\vec{P}	$\vec{V}' \times \vec{A}$ (appendix B)
P_w	power, w
p	semilatus rectum, m
q	dynamic pressure, $\frac{1}{2} \rho(V')^2$, newtons/m ²
R_r	radius of reference body, m
r	radius from origin to object, m
r_i	radius from origin to i^{th} perturbing body, m
S	aerodynamic reference area, m ²
T	temperature, °K
t	time, sec
U	gravitational potential
U_x, U_y, U_z	x,y,z accelerations due to gravity, m/sec ²
V	absolute velocity, m/sec
V'	relative velocity, m/sec
v	true anomaly, radians
X	forces acting on object other than gravity, thrust, lift, drag, and perturbations due to perturbing bodies
x,y,z	components of r , m
α	angle between thrust and velocity vectors (sketch (a)), deg
β	angle of rotation of thrust out of orbit plane (sketch (a)), deg
η	power efficiency factor
μ	$k^2 m_r$
ρ	atmospheric density, kg/m ³

ω argument of pericenter, radians

Ω equatorial longitude of ascending node, radians

Subscript:

0 initial value

APPENDIX B

VECTOR RESOLUTION

Relative Velocity

The relative velocity is defined as the velocity of the object with respect to the origin body. If the origin body is assumed to rotate about the z -axis, this velocity is given by

$$\vec{V}' = \vec{V} - \vec{\omega} \times \vec{r} \quad (B1)$$

In x,y,z component form,

$$V'_x = V_x + \omega y \quad (B2a)$$

$$V'_y = V_y - \omega x \quad (B2b)$$

$$V'_z = V_z \quad (B2c)$$

In the following sections, the atmosphere of the origin body is assumed to rotate as a solid body at the rate $\vec{\omega}$.

Thrust Resolution Along x,y,z Axes

The thrust direction is specified with respect to the relative velocity vector \vec{V}' by the angles α and β , as shown in sketch (a) (p. 4). For resolution of thrust vector into x,y,z components, it is convenient to define vectors \vec{A} and \vec{P} normal to and within the r, \vec{V}' plane, respectively, such that \vec{V}' , \vec{A} , and \vec{P} form an orthogonal set. Thus,

$$\vec{A} \equiv \vec{r} \times \vec{V}' = \text{relative angular momentum per unit mass} \quad (B3)$$

$$\vec{P} \equiv \vec{V}' \times \vec{A} \quad (B4)$$

The thrust vector can then be resolved in the $\vec{V}', \vec{A}, \vec{P}$ set as:

$$\vec{F} \cdot \vec{V}' = FV' \cos \alpha \quad (B5a)$$

$$\vec{F} \cdot \vec{A} = FA \sin \alpha \sin \beta \quad (B5b)$$

$$\vec{F} \cdot \vec{P} = FP \sin \alpha \cos \beta \quad (B5c)$$

Solving for \vec{F} yields

$$\vec{F} = \frac{F}{P^2} (V' \cos \alpha \vec{A} \times \vec{P} + A \sin \alpha \sin \beta \vec{P} \times \vec{V}' + P \sin \alpha \cos \beta \vec{P}) \quad (B6)$$

or, in x,y,z component form,

$$F_x = \frac{F}{p^2} \left[V' \cos \alpha (A_y P_z - A_z P_y) + A \sin \alpha \sin \beta (P_y V'_z - P_z V'_y) + P \sin \alpha \cos \beta P_x \right] \quad (B7a)$$

$$F_y = \frac{F}{p^2} \left[V' \cos \alpha (A_z P_x - A_x P_z) + A \sin \alpha \sin \beta (P_z V'_x - P_x V'_z) + P \sin \alpha \cos \beta P_y \right] \quad (B7b)$$

$$F_z = \frac{F}{p^2} \left[V' \cos \alpha (A_x P_y - A_y P_x) + A \sin \alpha \sin \beta (P_x V'_y - P_y V'_x) + P \sin \alpha \cos \beta P_z \right] \quad (B7c)$$

Aerodynamic Lift and Drag Resolution Along x,y,z Axes

The drag vector \vec{D} is aligned with the relative velocity vector \vec{V}' and is therefore given in x,y,z components as

$$\vec{D} = -D \frac{V'_x}{V'} - D \frac{V'_y}{V'} - D \frac{V'_z}{V'} \quad (B8)$$

The lift vector \vec{L} may be resolved into components along the previously defined orthogonal set \vec{V}' , \vec{A} , and \vec{P} by the following relations:

$$\vec{L} \cdot \vec{V}' = 0 \quad (B9a)$$

$$\vec{L} \cdot \vec{A} = LA \sin \beta \quad (B9b)$$

$$\vec{L} \cdot \vec{P} = LP \cos \beta \quad (B9c)$$

Solving for \vec{L} yields

$$\vec{L} = \frac{L}{p^2} (A \sin \beta \vec{P} \times \vec{V}' + P \cos \beta \vec{P}) \quad (B10)$$

or, in x,y,z component form,

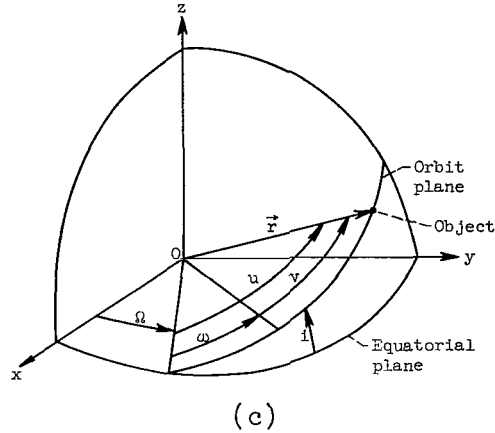
$$L_x = \frac{L}{p^2} \left[A \sin \beta (P_y V'_z - P_z V'_y) + P \cos \beta P_x \right] \quad (B11a)$$

$$L_y = \frac{L}{p^2} \left[A \sin \beta (P_z V'_x - P_x V'_z) + P \cos \beta P_y \right] \quad (B11b)$$

$$L_z = \frac{L}{p^2} \left[A \sin \beta (P_x V'_y - P_y V'_x) + P \cos \beta P_z \right] \quad (B11c)$$

APPENDIX C

TRANSFORMATION EQUATIONS FROM ORBIT ELEMENTS TO RECTANGULAR COORDINATES



From spherical trigonometry used in reference to the celestial sphere shown in sketch (c), the following relations may be derived for the position coordinates:

$$x = r(\cos \Omega \cos u - \sin \Omega \sin u \cos i) \quad (C1a)$$

$$y = r(\sin \Omega \cos u + \cos \Omega \sin u \cos i) \quad (C1b)$$

$$z = r(\sin u \sin i) \quad (C1c)$$

where

$$r = \frac{p}{1 + e \cos v} \quad (C2a)$$

$$u = \omega + v \quad (C2b)$$

and v can be obtained from

$$\cos v = \frac{\cos E - e}{1 - e \cos E} \quad (C2c)$$

and

$$M = E - e \sin E \quad (C2d)$$

The velocity components may be obtained by differentiating the position equations using the two-body relations $\dot{u} = \dot{v} = \frac{\sqrt{\mu p}}{r^2}$ and $\dot{r} = \sqrt{\frac{\mu}{p}} e \sin v$:

$$\dot{x} = -\sqrt{\frac{\mu}{p}} (N \cos i \sin \Omega + Q \cos \Omega) \quad (C3a)$$

$$\dot{y} = \sqrt{\frac{\mu}{p}} (N \cos i \cos \Omega - Q \sin \Omega) \quad (C3b)$$

$$\dot{z} = \sqrt{\frac{\mu}{p}} (N \sin i) \quad (C3c)$$

where

$$N = e \cos \omega + \cos u \quad (C4a)$$

$$Q = e \sin \omega + \sin u \quad (C4b)$$

APPENDIX D

RUNGE-KUTTA AND LOW-ORDER INTEGRATION

SCHEMES WITH ERROR CONTROL

The Runge-Kutta formula used is of fourth-order accuracy in step size h . It is of the form

$$X \Big|_1^2 \equiv X_2 - X_1 = \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (D1)$$

where

X = a dependent variable

$$X \Big|_1^2 = \text{increment in the dependent variable}$$

h_2 = increment in the independent variable t

$$k_1 = h_2 \dot{X}_2(t_1, X_1)$$

$$k_2 = h_2 \dot{X}_2 \left(t_1 + \frac{h_2}{2}, X_1 + \frac{k_1}{2} \right)$$

$$k_3 = h_2 \dot{X}_2 \left(t_1 + \frac{h_2}{2}, X_1 + \frac{k_2}{2} \right)$$

$$k_4 = h_2 \dot{X}_2(t_1 + h_2, X_1 + k_3)$$

A lower-order formula may be found by utilizing the three derivatives at $t = t_0, t_1$, and t_2 . If $h_1 = t_1 - t_0$ and $h_2 = t_2 - t_1$, the following Lagrangian interpolation formula gives the derivative at any time $t_0 \leq t \leq t_2$:

$$\dot{X} \equiv \dot{X}_0 \frac{(t - t_1)(t - t_2)}{h_1(h_1 + h_2)} - \dot{X}_1 \frac{(t - t_0)(t - t_2)}{h_1 h_2} + \dot{X}_2 \frac{(t - t_0)(t - t_1)}{h_2(h_1 + h_2)} \quad (D2)$$

Integration of this equation from t_1 to t_2 yields

$$X' \Big|_1^2 = \frac{1}{6} \left[\left(\frac{h_2}{h_1} \right)^2 \left(\frac{-h_2}{1 + \frac{h_2}{h_1}} \right) \dot{X}_0 + \frac{h_2}{h_1} (h_2 + 3h_1) \dot{X}_1 + \left(2h_2 + \frac{h_2}{1 + \frac{h_2}{h_1}} \right) \dot{X}_2 \right] \quad (D3)$$

The difference in the increments over the interval h_2 between the Runge-Kutta scheme and the low-order scheme may be divided by a nominal value of the dependent variable \bar{X} to obtain the relative error δ_2 . Thus,

$$\delta_2 = \left| \frac{X_1'^2 - X_1^2}{\bar{X}} \right| \quad (D4)$$

The error is expected to vary as approximately the fifth power of h , which leads to

$$\delta = Ah^5 \quad (D5a)$$

(where A is a suitable coefficient) or in the logarithmic form

$$\log \delta = A' + 5 \log h \quad (D5b)$$

where

$$A' = \log A \quad (D6a)$$

Let it be assumed that A' will vary linearly with t , the variable of integration. Then A' at a time corresponding to t_3 can be found from A' at two previous points t_1 and t_2 as

$$A'_3 = A'_2 + \frac{A'_2 - A'_1}{t_2 - t_1} (t_3 - t_2) \quad (D6b)$$

and if $h_3 = (t_3 - t_2)$ and $h_2 = (t_2 - t_1)$,

$$A'_3 = A'_2 + (A'_2 - A'_1) \frac{h_3}{h_2} \quad (D6c)$$

and on this basis δ_3 would be predicted to be

$$\log \delta_3 = A'_3 + 5 \log h_3 \quad (D7)$$

It is desired that δ_3 should approximate $\bar{\delta}$, the reference error; therefore,

$$\log h_3 = \frac{1}{5} (\log \bar{\delta} - A'_3) \quad (D8)$$

Each dependent variable has an associated relative error and would lead to computation of a different step size for each variable; however, the maximum relative error of all variables may be selected for $\bar{\delta}$. Obviously, inaccurate predictions of step size can occur when the maximum relative error shifts from one variable to another or when any sudden change occurs. When a step size produces

an excessively large error ($\delta > \delta_{\text{limit}}$), a reduced step size must be used. It may be obtained from the reference error $\bar{\delta}$ as

$$h_3 = \exp\left[\frac{1}{5} (\log \bar{\delta} - A_2^1)\right] \quad (\text{D9})$$

Starting the integration. - The Runge-Kutta scheme is simple to start, since integration from X_n to X_{n+1} requires no knowledge of X less than X_n . Since the error control coefficient A has no value at $t = 0$, a prediction of the second step size is difficult. To overcome this difficulty, two equal size first steps may be made before checking the error. The A for the first step may be arbitrarily set equal to the A for the second step so that h_3 may be predicted. The low-order integration scheme equation in this case becomes, with $h_2 = h_1$,

$$\left. X' \right|_1^2 = \frac{h_1}{3} (\dot{X}_0 + 4\dot{X}_1 + \dot{X}_2) \quad (\text{D10})$$

Failures. - Should two consecutive predictions of the same step fail to produce an error δ less than δ_{limit} , a return to the starting procedure will be made with a third prediction on step size, which is no larger than one-half of the second estimate. The step-size control described here will operate stably with nearly constant error per step only for a well-behaved function. For most problems it will repeat a step occasionally to reduce a large error, and on sharp corners it will restart. This action is not regarded as objectionable. The objective is to attain a desired level of accuracy with a minimum total number of steps.

APPENDIX E

GLOSSARY OF VARIABLES

VARIABLE	COMMON LOCATION	DEFINITION
A (700)	C(11)	ARRAY CONTAINING THE INITIAL DATA AND THE PROGRAM CONTROL VARIABLES
A1	B(10)	ERROR CONTROL PARAMETER DEFINED BY EQ. (D6A) AT T(1)
A2	B(11)	ERROR CONTROL PARAMETER DEFINED BY EQ. (D6A) AT T(2)
ACOE1	B(12)	INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)
ACOE2	B(13)	INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)
ACOE3	B(14)	INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)
AEXIT1 (10)	A(103)	ENGINE EXIT AREAS FOR AT MOST 10 STAGES, M**2
AEXIT	B(3)	AEXIT1(NSTAGE)
AK (3)	A(51)	RUNGE KUTTA COEFFICIENTS, SET IN STDATA
ALPHA	A(49)	ANGLE BETWEEN VELOCITY AND THRUST VECTORS, SEE SKETCH (A)
ALT	A(4)	VEHICLE ALTITUDE ABOVE EARTH, M
AM	B(90)	TOTAL VEHICLE ANGULAR MOMENTUM PER UNIT MASS, M**2/SEC
AMASS (30)	A(347)	PERMANENT LIST OF BODY MASSES IN ORDER OF PNAME LIST, SET IN STDATA, MASSES FROM ELIPS DATA BEGIN AT AMASS (21), SUN MASS UNITS
AMC (3)	B(87)	X,Y,Z COMPONENTS OF ANGULAR MOMENTUM PER UNIT MASS, **2)/SEC
AMSQRD	B(91)	SQUARE OF TOTAL ANGULAR MOMENTUM PER UNIT MASS, M**4/SEC**2
AREA1 (10)	A(113)	AERODYNAMIC REFERENCE AREAS FOR AT MOST 10 STAGES, M**2
AREA	B(6)	AREA1(NSTAGE)
ASYMPT	A(7)	SEE TABLE II
ATMN	A(21)	SEE TABLE II
AU	A(29)	ASTRONOMICAL UNIT, M
AW (4)	A(55)	RUNGE KUTTA COEFFICIENTS, SET IN STDATA
AZI	A(35)	INITIAL AZIMUTH ANGLE, USED WHEN IMODE = 4, SEE SKETCH (B), DEGREES

B (800)	C(1111)	ARRAY CONTAINING INTERNAL PARAMETERS NOT UNDER USER CONTROL
BETA	A(50)	ANGLE BETWEEN VELOCITY-THRUST PLANE AND ORBIT PLANE, SEE SKETCH(A)
BMASS (8)	B(137)	BODY MASSES SELECTED FROM AMASS LIST IN SEQUENCE CORRESPONDING TO BNAME LIST
BNAME (8)	B(122)	ORDERED LIST OF BCD BODY NAMES
BODYCD (10)	A(143)	ORIGINAL UNORDERED LIST OF BCD BODY NAMES READ IN AT INPUT
BODYL (10)	B(153)	AUXILIARY ORDERED LIST OF BCD BODY NAMES
CD	A(165)	TOTAL DRAG COEFFICIENT
CDI	A(163)	INDUCED DRAG COEFFICIENT
CHAMP	B(25)	SMALLEST CRITICAL RADIUS (RBCRIT(J)) WITHIN WHICH OBJECT LIES
CINCL	B(55)	COSINE OF INCLINATION
CIRCUM	B(82)	CIRCUMFERENTIAL COMPONENT OF TOTAL PERTURBATIVE ACCELERATION, M/SEC**2
CL	A(164)	LIFT COEFFICIENT
CLEAR	C(3)	SEE TABLE II
COEFN (192)	A(407)	STORAGE ARRAY FOR COEFFICIENTS USED TO COMPUTE ALPHA, CL, CDI, CD OR OTHER PARAMETERS
COMPA (3)	B(63)	COMPONENTS OF TOTAL PERTURBATIVE ACCELERATION ALONG X, Y, Z AXES.
CONSU	A(31)	SEE TABLE II
CONSTU	A(32)	SEE TABLE II
COSALF	B(48)	COSINE OF ALPHA
COSBET	B(49)	COSINE OF BETA
COSTRU	B(53)	COSINE OF TRU
COSV	B(57)	COSINE OF THE ARGUMENT OF LATITUDE
D (1100)	C(2111)	ARRAY WHERE SAVED DATA IS STORED FOR LATER USE. ARRAYS A, XPRIM, AND XPRIMB MAY BE SAVED.
DELMAX	A(19)	SEE TABLE II
DEL	A(43)	OUTPUT CONTROL PARAMETER USED IN STEP
DELT1 (10)	A(133)	INITIAL STEP SIZES FOR AT MOST 10 STAGES, SEC
DELT	B(1)	DELT1(NSTAGE)
DENSITY	B(29)	ATMOSPHERIC DENSITY, KG/M**3
DONE	B(39)	CONTROL PARAMETER FROM STEP WHICH INFORMS NBODY TO STOP INTEGRATING

DRAG (3)	B(69)	X,Y,Z COMPONENTS OF THE DRAG ACCELERATION,M/SEC**2
DTOFFJ	A(23)	JULIAN DATE OF TAKEOFF
E2	B(18)	LARGEST OF THE RELATIVE ERRORS BETWEEN R-K AND LOW-ORDER INTEGRATION METHODS,EQ. (D4)
EFMRS (7)	B(130)	LIST OF BCD BODY NAMES WHOSE POSITIONS ARE TO BE DETERMINED FROM TAPE DATA
ELEV	A(36)	INITIAL ELEVATION ANGLE,USED WHEN IMODE=4, SKETCH(B), DEGREES
ELIPS (12,10)	A(167)	ELLIPSE DATA FOR PERTURBATING BODIES,READ FROM CARDS, 12 PIECES OF DATA PER BODY
EMONE	B(28)	ECCENTRICITY -1
END	A(5)	SEE TABLE II
EPAR	B(26)	SQUARE ROOT OF (ECCENTRICITY SQUARED -1)
EREF	A(13)	SEE TABLE II
ERLIMT	A(14)	SEE TABLE II
ERLOG	B(17)	NATURAL LOGARITHM OF EREF
ETOL	A(30)	SEE TABLE II
EXITA	B(392)	AEXIT(NSTAGE)/100, NEWTONS/MB
EXMODE	B(27)	ECCENTRICITY CALCULATED WHEN IMODE=3
FILE	B(22)	SEE TABLE II
FLOW1 (10)	A(83)	RATE OF PROPELLENT FLOW, KG/SEC
FLOW	B(5)	FLOW1(NSTAGE)
FORCE (3)	B(66)	X,Y,Z COMPONENTS OF THRUST ACCELERATION, M/SEC**2
GASFAC	A(46)	DEFINED IN SUBROUTINE AERO, SET IN STDATA
GEOH	B(32)	GEOPOTENTIAL, M
GK2M	B(36)	GRAVITATIONAL CONSTANT,MU,OF THE SYSTEM,M**3/SEC**2
GKM	B(37)	SQUARE ROOT OF GK2M
H2	B(15)	VALUE OF DELT FOR PREVIOUS STEP
I8ODY (8)	B(177)	DEFINED IN SUBROUTINE ORDER
ICC (10)	A(153)	SEE TABLE II
IDENT (10)	A(123)	INPUT IDENTIFICATION NUMBERS ASSOCIATED WITH EACH STAGE
IMODE	A(1)	SEE TABLE II
IND (3)	A(60)	SET OF INDICES, SET IN STDATA
INDERR	B(51)	NUMBER OF SETS OF ERROR DATA, SET IN ERRORZ FOR USE IN N8ODY

INLOOK	A(599)	INPUT IDENTIFICATION NUMBER FOR INPUT AFTER FINDING C (LOOKX) = XLOOK
KSUB	B(19)	INDEX OF RUNGE-KUTTA SUBINTERVALS
LAT	A(33)	INITIAL GEODCENTRIC LATITUDE, USED WHEN IMODE=4, SKETCH (8), DEGREES
LONG	A(34)	INITIAL LONGITUDE RELATIVE TO GREENWICH, USED WHEN IMODE=4, SKETCH(8), DEGREES
LOOKX	A(8)	SEE TABLE II
LOOKSW	A(9)	SEE TABLE II
LSTAGE	A(38)	TOTAL NUMBER OF STAGES INTEGRATED BEFORE RETURNING TO THE MAIN PROGRAM
MBODY5	B(42)	NUMBER OF PERTURBATING BODIES (NBODY5-1)
MODOUT	A(20)	SEE TABLE II
NBODY5	B(41)	TOTAL NUMBER OF BODIES, EXCLUDING THE VEHICLE
NCASE5	A(600)	SAVED VALUE OF NCASE
NCASE	C(1)	CASE NUMBER, RAISED ONCE EACH TIME CONTROL PASSES THROUGH THE MAIN PROGRAM
NEFMRS (8)	B(185)	DEFINED IN SUBROUTINE ORDER
NEQ	A(2)	NUMBER OF EQUATIONS TO BE INTEGRATED, SET TO 8 IN STDATA
NSAVE	C(4)	SEE TABLE II
NSTAGE	A(3)	THE INDEX INDICATING THE PARTICULAR STAGE CURRENTLY BEING INTEGRATED
NSTART	B(24)	INTERNAL CONTROL IN NBODY AND EQUATE
OBLATJ	A(26)	OBLATENESS COEFFICIENT OF SECOND HARMONIC
OBLATO	A(27)	OBLATENESS COEFFICIENT OF FOURTH HARMONIC
OBLATH	A(28)	OBLATENESS COEFFICIENT OF THIRD HARMONIC
OBLATN	A(40)	SEE TABLE II
OBLAT (3)	B(75)	X,Y,Z COMPONENTS OF OBLATENESS ACCELERATION, M/SEC**2
OLDDEL	B(9)	VALUE OF DELT FOR PREVIOUS GOOD STEP
ORBELS (6)	B(116)	ARRAY OF OUTPUT VARIABLES, EITHER RECTANGULAR OR ORBIT ELEMENTS
OUTPOT	B(399)	CAUSES ABSENCE OF OUTPUT WHEN NONZERO
P (3)	B(84)	DEFINED IN EQ. (B4)

PAR (3)	B(60)	DEFINED BY EQUATIONS IN SUBROUTINE THRUST
PMAGN	B(50)	DEFINED IN EQUATION FORM BY SUBROUTINE THRUST
PNAME (30)	A(287)	PERMANENT LIST OF BODY NAMES MADE FROM PNAALIST IN SUBROUTINE ORDER, ELIPS NAMES BEGIN AT PNAME(21)
PRESS	B(33)	ATMOSPHERIC PRESSURE, MB
PSI	B(30)	PATH ANGLE, ANGLE BETWEEN PATH AND LOCAL HORIZONTAL, DEGREES
PSIR	B(398)	RELATIVE PATH ANGLE, TAKEN RELATIVE TO A ROTATING ORIGIN BODY, DEG
PUSH	A(166)	THRUST FORCE, NEWTONS
PUSHO	B(391)	VACUUM THRUST FORCE, NEWTONS
Q	B(59)	DYNAMIC PRESSURE, NEWTONS/M**2
QMAX	B(44)	MAXIMUM VALUE OF Q DEVELOPED DURING A SINGLE TRAJECTORY (SET TO ZERO WHEN CONTROL PASSES THROUGH SUBROUTINE EXTRA)
QX (3)	B(78)	X,Y,Z COMPONENTS OF PERTURBATIVE ACCELERATION DUE TO PERTURBATING BODIES, M/SEC**2
R (8)	B(102)	DISTANCES OF ALL BODIES FROM OBJECT, IN ORDER OF BNAME LIST, M
RADIAL	B(81)	RADIAL COMPONENT OF TOTAL PERTURBATIVE ACCELERATION, POSITIVE OUTWARD, M/SEC**2
RAMC (5)	B(393)	RELATIVE ANGULAR MOMENTUM PER UNIT MASS COMPONENTS, TOTAL RELATIVE ANGULAR MOMENTUM PER UNIT MASS, AND ITS SQUARE, M**2/SEC
RATM	A(22)	RADIUS OF ATMOSPHERE, M
RATMOS	B(23)	SET EQUAL TO RATM WHEN ATMN EQUALS THE REFERENCE BODY NAME, BNAME(1)
RATIO	B(58)	RATIO OF ADJACENT STEP SIZES, DELT
RB (3,8)	B(193)	X,Y,Z COMPONENTS OF DISTANCE FROM ALL BODIES TO THE OBJECT, M
RBCRIT (8)	B(145)	LIST OF SPHERE-OF-INFLUENCE RADII OF ALL BODIES IN BNAME LIST, M
RCRIT (30)	A(377)	PERMANENT LIST OF SPHERE-OF-INFLUENCE RADII CORRESPONDING TO PNAME LIST OF BODY NAMES. RADII FROM ELIPS DATA BEGIN AT RCRIT(21), M
RE	A(25)	RADIUS OF EARTH EQUATOR, M
RECALL	C(5)	SEE TABLE II
REFER (30)	A(317)	LIST OF REFERENCE BODIES CORRESPONDING TO PNAME LIST, REFERENCE BODIES FROM ELIPS DATA BEGIN AT REFER(21)
RESQRD	B(7)	SQUARE OF RE
RETURN	B(400)	CAUSES CONTROL NOT TO RETURN TO MAIN PROG. IF NONZERO

REVS	A(48)	REVOLUTION COUNTER, USED ONLY FOR OUTPUT
REVOLV	B(21)	ROTATION RATE OF REFERENCE BODY WHEN ATMN=BNAME(1), RAD/SEC
RMASST (10)	A(73)	INITIAL MASSES FOR AT MOST 10 STAGES,KG
ROTATE	A(39)	ROTATION RATE OF A REFERENCE BODY, RAD/SEC
RSQRD	B(45)	RADIUS SQUARED OF OBJECT TO ORIGIN, M**2
SIGNAL	B(31)	SEE TABLE II
SIMP1 (10)	A(93)	SPECIFIC IMPULSES FOR AT MOST 10 STAGES, SEC
SIMP	B(2)	SIMP1(NSTAGE)
SINALF	B(46)	SINE OF ALPHA
SINBET	B(47)	SINE OF BETA
SINTRU	B(52)	SINE OF TRU
SINCL	B(54)	SINE OF INCLINATION
SINV	B(56)	SINE OF THE ARGUMENT OF LATITUDE
SPACES	B(16)	NUMBER OF EQUAL TIME UNITS UNTIL NEXT OUTPUT
SPD	A(44)	SECONDS PER DAY, SET IN STDATA, SEC/DAY
SQRDK1	A(47)	GRAVITATIONAL CONSTANT OF THE SUN, AU**3/DAY**2
SQRDK	B(35)	GRAVITATIONAL CONSTANT OF THE SUN, M**3/SEC**2
STEPMX	A(16)	SEE TABLE II
STEPS	A(17)	SEE TABLE II
STEP60	A(41)	COUNT OF SUCCESSFUL INTEGRATION STEPS
STEPND	A(42)	COUNT OF UNSUCCESSFUL INTEGRATION STEPS (THOSE WHICH DO NOT PASS ERROR CONTROL TEST)
SWLOOK	A(10)	SEE TABLE II
TABLT	B(20)	TIME MEASURED RELATIVE TO THE JULIAN DATE OF TAKEOFF, DAYS
TABLE (200)	C(1911)	ARRAY OF INPUT PARAMETERS AND THEIR COMMON STORE LO- CATIONS
TAPE3	C(2)	SEE TABLE II
TB (10)	A(63)	FLIGHT TIMES FOR AT MOST 10 STAGES, SEC

TDATA (6,3,7)	B(265)	COEFFICIENTS FROM EPHEMERIDES TAPE TO BE USED IN DETERMINING POSITIONS AND POSSIBLY VELOCITIES OF PERTURBATING BODIES, ONE SET FOR EACH OF 7 BODIES
TDEL (7)	B(170)	ONE-HALF OF TIME SPACING BETWEEN TWO ADJACENT ENTRIES OF LIKE BODY NAME ON EPHEMERIDES TAPE, READ FROM TAPE FOR EACH BODY
TFILE	A(6)	SEE TABLE II
TIM (7)	B(163)	TIME FOR SET OF EPHEMERIS DATA, READ FROM EPHEMERIDES TAPE, ONE FOR EACH BODY
TKICK	A(15)	INITIAL STEP SIZE OF A TRAJECTORY TO BE COMPUTED IN CLOSED-FORM, FOR USE WHEN IMODE=4, WHICH FACILITATES STARTING OF SOME TYPES OF TRAJECTORIES
TM	B(34)	ATMOSPHERIC TEMPERATURE TIMES THE RATIO OF MOLECULAR TO ACTUAL MOLECULAR WEIGHT, DEGREES KELVIN
TMAX	B(4)	SEE TABLE II
TMIN	A(18)	SEE TABLE II
TOFFT	A(24)	FRACTIONAL PART OF JULIAN DATE OF TAKEOFF, DAYS
TRSFER	B(8)	SEE TABLE II
TRU	B(40)	TRUE ANOMALY, RAD
TTEST	A(54)	SEE TABLE II
TTOL	A(45)	TIME TOLERANCE WITHIN WHICH PROBLEM TIME MINUS TMAX MUST LIE TO END STAGE
U	A(59)	ECCENTRIC ANOMALY, RAD
V	B(95)	VELOCITY OF OBJECT RELATIVE TO THE ORIGIN, M/SEC
VATM (3)	B(97)	X,Y,Z COMPONENTS OF THE RELATIVE VELOCITY, VQ,M/SEC
VEFM (3,8)	B(241)	X,Y,Z COMPONENTS OF OBJECT VELOCITY RELATIVE TO ALL BODIES, M/SEC
VEL	A(37)	INITIAL RELATIVE VELOCITY, USED WHEN IMODE=4, SKETCH (B), M/SEC
VMACH	B(38)	MACH NUMBER OF OBJECT
VQ	B(100)	VELOCITY OF OBJECT RELATIVE TO ATMOSPHERE, M/SEC
VQSQRD	B(101)	SQUARE OF VQ, M**2/SEC**2
VSQRD	B(96)	SQUARE OF V, M**2/SEC**2
VX	B(92)	X COMPONENT OF VELOCITY, M/SEC
VY	B(93)	Y COMPONENT OF VELOCITY, M/SEC
VZ	B(94)	Z COMPONENT OF VELOCITY, M/SEC
X (100)	B(401)	WORKING SET OF INTEGRATION VARIABLES
XDOT (100)	B(501)	TIME DERIVATIVES OF THE SET X

XIFT (3)	B(72)	X,Y,Z COMPONENTS OF LIFT ACCELERATION, M/SEC**2
XINC (100)	B(601)	INCREMENTS OF THE INTEGRATION VARIABLES PER STEP
XLOOK	A(12)	SEE TABLE II
XP (3,8)	B(217)	X,Y,Z COMPONENTS OF PERTURBATING BODY POSITIONS RELATIVE TO ORIGIN
XPRIM (100,2)	C(711)	TWO 100-ELEMENT SETS, THE FIRST SET CONTAINS VALUES OF THE INTEGRATION VARIABLES AT THE PREVIOUS GOOD STEP, THE SECOND SET IS UNDER THE INTEGRATION PROCESS, SEE TABLE V
XPRIMB (100,2)	C(911)	LEAST SIGNIFICANT HALF OF DOUBLE PRECISION INTEGRATION VARIABLES XPRIM
XTOL	A(11)	TOLERANCE ON THE DISCIMINATION C(LOOKX)-XLOOK TO BE SATISFIED
XWHOLE (6)	B(110)	RECTANGULAR COORDINATES AND VELOCITIES, SET ASIDE FOR USE IN ORIGIN TRANSLATIONS
ZN	B(43)	MEAN ANGULAR MOTION OF OBJECT, RAD/SEC
ZORMAL	B(83)	Z COMPONENT OF TOTAL PERTURBATIVE ACCELERATION, M/SEC**2

APPENDIX F

LEWIS RESEARCH CENTER EPHEMERIS

General Description

The ephemeris data initially available on magnetic tape were from the Themis code prepared by the Livermore Laboratory, evidently from U.S. Naval Observatory data. Later, an ephemeris was obtained from the Jet Propulsion Laboratory assembled as a joint project of the Jet Propulsion Laboratory and the Space Technology Laboratory. These data are given relative to the mean vernal equinox and equator of 1950.0 and are tabulated with ephemeris time as the argument.

An ephemeris was desired for certain uses in connection with the IBM 7090 computer that would be shorter than the original ephemeris tapes mentioned and would be as accurate as possible consistent with the length. A short investigation of the various possibilities led to adoption of fitted equations. In particular, fifth-order polynomials were simultaneously fitted to the position and velocities of a body at three points. This procedure provides continuity of position and velocity from one fit to the next, because the exterior points are common to adjacent fits. Polynomials were selected rather than another type of function, because they are easy to evaluate. Three separate polynomials are used for the x , y , and z coordinates, respectively.

Procedure Used to Fit Data

The process of computing the fitting equations is as follows:

(1) A group of 50 sets of the components of planetary position was read into the machine memory for a single planet together with differences as they existed on the original magnetic tape. The differences were verified by computation (in double precision because some data required it); and any errors were investigated, corrected, and verified. Published ephemeris data were adequate to correct all errors found.

(2) The components of velocity v_x , v_y , and v_z were computed and stored in the memory for each of the 50 positions by means of a numerical differentiation formula using ninth differences; namely,

$$\dot{X} = (T_1 - T_{-1}) \left[\frac{\Delta I_{-1} + \Delta I_{+1}}{2} - \frac{\Delta III_{-1} + \Delta III_{+1}}{12} + \frac{\Delta V_{-1} + \Delta V_{+1}}{60} - \frac{\Delta VII_{-1} + \Delta VII_{+1}}{280} + \frac{\Delta IX_{-1} + \Delta IX_{+1}}{1260} \right] \quad (F1)$$

(See ref. 11, pp. 42 and 99 for notation.) Double-precision arithmetic was used for differences, but velocities were tabulated with single precision.

(3) Coefficients C, D, E, and F in the fifth-order polynomial

$$X = X_0 + \dot{X}_0(T - T_0) + C(T - T_0)^2 + D(T - T_0)^3 + E(T - T_0)^4 + F(T - T_0)^5 \quad (F2)$$

and its derivative

$$\dot{X} = \dot{X}_0 + 2C(T - T_0) + 3D(T - T_0)^2 + 4E(T - T_0)^3 + 5F(T - T_0)^4 \quad (F3)$$

were found to fit a first point (which was far enough from the beginning point to have all differences computed) and two equally spaced points for each component of position and velocity. (The initial spacing is not important, as will be seen later.) Spacing is defined as the number of original data points fitted by one equation. Single-precision arithmetic was used.

(4) The coefficients C, D, E, and F in step (3) were then used in equations (F2) and (F3) to calculate components of all positions and velocities given in the original data and lying within the interval fitted. These values were checked with the original data. Radius R and velocity V were computed at the times tabulated in the original data. If any component of the position differed from the original data by more than $R \times 10^{-7}$ or if any velocity differed from the original by more than $V \times 10^{-6}$, the fit was considered unsatisfactory.

(5) If the fit was considered unsatisfactory, this fact was recorded, and the spacing was reduced by two data points. Steps 2 to 4 were then repeated. If the fit was considered satisfactory, this fact was recorded, and the spacing was increased by two spaces. Steps 2 to 4 were repeated. The largest satisfactory fit was identified when a certain spacing was satisfactory and the next larger fit was not satisfactory.

(6) The coefficients that corresponded to the largest satisfactory fit were recorded on tape in binary mode as follows:

Word number	Data	Mode	Definitions and/or units
1	Planet name	BCD	Six characters (first six)
2	Julian date	Floating point	Date of midpoint of fit, Julian date
3	Delta T		Number of days on each side of midpoint
4	F _x		aAU/day ⁵
5	E _x		aAU/day ⁴
6	D _x		aAU/day ³
7	C _x		aAU/day ²
8	\dot{x}		aAU/day
9	x		aAU
10	F _y		aAU/day ⁵
11	E _y		aAU/day ⁴
12	D _y		aAU/day ³
13	C _y		aAU/day ²
14	\dot{y}		aAU/day
15	y		aAU
16	F _z		aAU/day ⁵
17	E _z		aAU/day ⁴
18	D _z		aAU/day ³
19	C _z		aAU/day ²
20	\dot{z}		aAU/day
21	z		aAU

^aExcept for Moon data, which are in Earth radii and days.

(7) As soon as a set of coefficients was selected for an interval, additional data were read from the source ephemeris tape and used to replace the points already fitted (except the last point). These data were processed as described in steps 1 and 2 so that the next 50 points were ready to be fitted. Steps 3 to 6 were then used to find the next set of coefficients, and steps 1 to 6 were repeated until all data for all planets were fitted.

Data Treated

The preceding process was applied to all data available at the time. For the Moon, the technique usually led to the use of every point in the fitted interval (i.e., only three points were fitted). Thus, a check of accuracy was not available. The error in the attempt to fit the next greater interval (five points) was not excessive, however, and it is judged that the accuracy obtained from these fits is about equal to that held on the other bodies.

Merged Ephemeris Tape

Once all the positions and velocities of all the bodies then available were fitted, the coefficients were merged in order of the starting date of each fit. The resulting tape was written in binary mode with 12 sets of fits per record.

The detail of this record is as follows:

	1st word:	FORTTRAN compatible
	2nd word:	file number, fixed point in decrement
	3rd word:	planet name, code in BCD, first six characters
Set 1	4th word:	Julian date, floating point
	-	etc., according to list in paragraph 6
	-	21 words
	23rd word:	z
	24th word:	planet name, code in BCD, first six characters
Set 2	25th word:	Julian date, floating point
	-	
	-	
	44th word:	z

Successive sets follow one another with a total of 12 sets.

Set 12	234th word:	planet name
(last set)	235th word:	Julian date, floating point
	254th word:	z
	End-of-record gap	

One record contains 254 words, the first is for FORTRAN compatibility, the second is a file number used for identification in the system. It is a fixed point 2. The third is the beginning of the first set of data, and 12 sets follow each with 21 words. The last word is the 254th word (counting the FORTRAN compatible word)

followed by an end-of-record gap. The remaining records are compiled in the same manner with an end-of-file recorded as a terminating mark.

Because of the merging operation, all bodies are given in one list in a random order according to the starting date of the interval. The starting date is the Julian day (word 2) minus the half interval (word 3) (see procedure, paragraph 6). The entire ephemeris occupies about one-seventh reel of tape. A summary of data is given in table VII.

APPENDIX G

INPUT-DATA REQUIREMENTS

The procedure needed to run actual problems with the aid of this routine is described herein. It is intended to permit the user with a specific problem in mind to make a complete list of data required and to select desirable operating alternatives from those available. The details of this procedure are contained in the following instructions:

(1) Provision has been made for two types of ephemeris data to specify the locations of celestial bodies that perturb the vehicle. They are ellipse data and ephemeris-tape data. If the problem does not involve perturbing bodies (except a reference body) or if elliptic data are used for all the perturbing bodies, skip to instruction 5.

(2) If the perturbing-body data are to be taken from an ephemeris tape, list the names of the ephemerides and Julian dates to be covered along with the following auxiliary information:

1st card: \$DATA = 300, \$TABLE, 2 = TAPE 3, 17 = ELIST, 29 = TBEGIN,
30 = TEND/

Other cards: TAPE 3 = 0

TBEGIN = ephemeris beginning Julian date

TEND = ephemeris ending Julian date

ELIST = (names of perturbing bodies in "ALF" format, see example in text)

The ephemerides of all planets except Earth bear the name of the planet. The ephemeris giving the distance from Earth to the Sun is called "sun," as is astronomical practice.

(3) If successive files on the ephemeris tape are to be made, punch the corresponding sets as follows:

\$DATA = 300, TAPE 3 = 0, TBEGIN = , TEND = , ELIST =

As many similar sets as are needed may be appended.

(4) If ellipse data are to be loaded from cards, they are prepared later under instruction 11.

(5) On the first execution after loading the routine, the common area is cleared whether an ephemeris tape is constructed or not. It is now necessary to load a table of variable names. Once loaded, this table will not be cleared again (except if the control variable TAPE 3 is set equal to zero). These names are for use on the input cards. If a different name is desirable for any

variable, it may be changed in the table and where it appears on the input card (ref. 7). The cards are:

\$DATA=1, \$TABLE, 33=DTOFFJ, 34=TOFFT, 711=TIME, 716=X, 717=Y, 718=Z, 713=VX, 714=VY, 715=VZ, 11.=IMODE, 713=E, 714=OMEGA, 715=NODES, 716=INCL, 717=MA, 718=P, 43=LAT, 44=LONG, 45=AZI, 46=ELEV, 14=ALT, 47=VEL, 16=TFILE, 28=TMIN, 153=BODYCD, 177=ELIPS, 30.=MODOUT, 27=STEPS, 29=DELMAX, 26=STEPMX, 23=EREF, 24=ERLIMT, 4.=NSAVE, 5=RECALL, 3=CLEAR, 18.=LOOKX, 22=XLOOK, 19.=LOOKSW, 20=SWLOOK, 609.=INLOOK, 15=END, 31=ATMN, 32=RATM, 49=ROTATE, 417=COEFN, 163.=ICC, 60=BETA, 50=OBLATN, 73=TB, 93=FLOW, 103=SIMP, 123=AREA, 143=DELT, 83=RMASS, 113=AEXIT, 133.=IDENT, 48.=LSTAGE, 25=TKICK /

(6) The initial position and velocity of the vehicle may be given in any one of the three coordinate systems. If the initial data are given in orbit elements, skip to instruction (8). If the initial data are given in rectangular coordinates, skip to instruction (7). If the initial data are given in Earth-centered spherical coordinates, the following variables should be punched:

LAT = latitude, deg, positive north of equator

LONG = longitude, relative to Greenwich, deg

ALT = altitude above sea level, m

AZI = azimuth angle, east from north, deg

ELEV = elevation angle, horizontal to path, deg

VEL = initial relative velocity, m/sec

TKICK = size of initial vertical, nondrag step to facilitate starting, sec

If the Earth is assumed to be rotating but aerodynamic forces are not to be considered, set

ROTATE = Earth rotation rate, $7.29211585 \times 10^{-5}$ radian/sec

If integration in rectangular coordinates is desired set

IMODE = 4

or else if integration in orbit elements is desired set

IMODE = -4

Skip to instruction (9).

(7) If the initial data are in rectangular coordinates, set the following variables:

X = x-component of position in x,y,z coordinate system, m

Y = y-component of position in x,y,z coordinate system, m

Z = z-component of position in x,y,z coordinate system, m
 VX = x-component of velocity in x,y,z coordinate system, m/sec
 VY = y-component of velocity in x,y,z coordinate system, m/sec
 VZ = z-component of velocity in x,y,z coordinate system, m/sec
 If integration in rectangular coordinates is desired set
 $IMODE = 2$
 or else, if integration in orbit elements is desired set
 $IMODE = -2$

Skip to instruction (9).

- (8) If the initial data are in orbit elements, set the following variables:

E = eccentricity

$OMEGA$ = argument of pericenter, radians

$NODES$ = longitude of ascending node (to mean vernal equinox of 1950.0), radians

$INCL$ = orbit inclination to mean equator of 1950.0, radians

MA = mean anomaly, radians

P = semilatus rectum, m

If integration in orbit elements is desired set
 $IMODE = 1$
 or else, if integration in rectangular coordinates is desired set
 $IMODE = -1$

- (9) To specify takeoff time, set the following variables:

$DTOFFJ$ = Julian day number

$TOFFT$ = fraction of day

$TIME$ = time from previously set Julian date, sec

Takeoff occurs at the instant (ephemeris time) corresponding to the sum of the last three quantities. If a specific date or time is not required, these variables may be skipped. In that case, the SUBROUTINE STDATA sets $DTOFFJ$ to 2440 000.

- (10) To specify the origin and any perturbing bodies, list them as $BODYCD =$ (list of body names in "ALF" format, see text example). The first body in the list is taken to be the reference body. The distances between the bodies in

this list must be computable from either ellipse data (instruction (11)) or ephemeris-tape data (instruction (2)). There may be no more than eight names in the list. Also, if the ephemeris tape is being used, the correct file must be found on it. For this purpose, set `TFILE` = desired ephemeris-tape file. The ephemeris files were numbered in sequence when written in instruction (2). If `TFILE` is not given, it will be set equal to 1.0 by the SUBROUTINE `STDATA`.

(11) For each body whose path is represented by an ellipse, a 12-element set of data must be loaded. A 12-element set consists of:

1. Body name in "ALF" format (maximum of six characters)
2. Reference body name in "ALF" format (maximum of six characters)
3. Mass of body, sun mass units
4. Radius of sphere of influence, m
5. Semilatus rectum, AU
6. Eccentricity
7. Argument of pericenter, radians
8. Longitude of ascending node (to mean vernal equinox of 1950.0),
radians
9. Orbit inclination (to mean equator of 1950.0), radians
10. Julian day at perihelion
11. Fraction of day at perihelion
12. Period, mean solar days

It is convenient to punch a 12-element set in sequence and to separate the elements by commas on as many cards as are required. Several sets may then be loaded consecutively. The order of the sets is immaterial. Ellipse data, if present, take precedence over ephemeris-tape data. The sets are loaded consecutively, in any order, as follows:

`ELIPS` = set 1, set 2, set 3, . . . , set n ; $n \leq 10$ (see example in appendix I)

(12) If oblateness effects of the Earth are to be included, set

`OBLATN` = (ALF5)EARTH

(13) Provision has been made to fly multistage vehicles with up to 10 stages. At least one stage must be loaded. There are eight parameters for each stage with provision for input-controlled modifications of other variables. The 10 values of each parameter are stored in an array corresponding to the

10 stages. Input cards are as follows:

TB = burning time for 1st stage, 2nd stage, etc., sec

FLOW = propellant flow rate for 1st stage, 2nd stage, etc., kg/sec

SIMP = vacuum specific impulse of 1st stage, 2nd stage, etc., sec

AREA = aerodynamic reference area of 1st stage, 2nd stage, etc., m^2

AEXIT = engine exit area for 1st stage, 2nd stage, etc., m^2

RMASS = initial mass or jettison mass for 1st stage, 2nd stage, etc., kg

DELT = initial integration step size for 1st stage, 2nd stage, etc., sec

IDENT = input identification number 1st stage, 2nd stage, etc.

TB must be loaded for as many stages as are to be flown. Others may be omitted if zero is appropriate. If RMASS(i) is not positive, the ith stage begins with the final mass of the previous stage reduced by the fixed amount RMASS(i). In the case of DELT, zero will result in use of TB/100. IDENT of a nonzero value will cause any data cards of that identification number to be read in after the stage is set up and before integration begins. This permits the user to make almost any change desired. The order of data cards is discussed in instruction (24).

(14) The thrust orientation must be specified by setting

BETA = angle β , deg (see sketch (a) (p. 4))

COEFN (I) = angle-of-attack schedule, $\alpha = \alpha(t)$ (see instruction (16))

ICC = fixed-point integer (see instruction (16))

For the special case of tangential thrust, none of the last three variables need be set.

(15) If aerodynamic forces are present, set in addition to AREA in instruction (13):

ATMN = name of body that has atmosphere, in "ALF" format, (Earth)

RAIM = radius above which atmospheric forces are not to be considered, m

ROTATE = atmospheric-rotation rate, radians/sec ($7.29211585 \times 10^{-5}$ for Earth)

BETA = angle β , deg (see sketch (a))

COEFN (I) = angle-of-attack schedule, $\alpha = \alpha(t)$, $C_L/\sin \alpha$, $C_{D,0}$, and $C_{D,i}/C_L^2$ curves (see instruction (16))

ICC = fixed-point integers (see instruction (16))

(16) If neither thrust nor aerodynamic forces are present, skip to instruction (18). The relations $\alpha(t)$, $C_L/\sin \alpha$, $C_{D,0}$, and $C_{D,i}/C_L^2$ are assumed to be quadratic functions that involve coefficients, which are located in the COEFN(J) array. The arrangement of these coefficients is best explained by an example. Suppose the function $\alpha(t)$ is as follows:

$$\alpha = \begin{cases} a_{11} + a_{12}t + a_{13}t^2 & (t_1 \leq t \leq t_2) \\ a_{21} + a_{22}t + a_{23}t^2 & (t_2 \leq t \leq t_3) \\ a_{31} + a_{32}t + a_{33}t^2 & (t_3 \leq t \leq t_4) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \text{etc.} & \text{etc.} \end{cases}$$

The coefficients $a_{i,j}$ should then be loaded into the COEFN(J) array as:

$$\text{COEFN}(J) = t_1, a_{11}, a_{12}, a_{13}, t_2, a_{21}, a_{22}, a_{23}, t_3, a_{31}, a_{32}, a_{33}, t_4, \dots, t_n$$

Furthermore, additional sets of coefficients for the other functions may simply be added to the COEFN(J) array, which results in a string of sets of coefficients, and can be represented, for example, as:

$$\begin{aligned} \text{COEFN}(J) &= \alpha \text{ coefficients, } C_L/\sin \alpha \text{ coefficients, } C_{D,0} \text{ coefficients, etc.} \\ &= t_1, a_{11}, a_{12}, \dots, t_n, N_{M,1}, b_{11}, b_{12}, \dots, N_{M,k}, \text{ etc.} \end{aligned}$$

The starting point in the COEFN(J) array of each function must also be loaded to identify the correct region of coefficients. To this end, the following array must also be loaded:

ICC(1) = fixed-point value of J where α coefficients begin

ICC(2) = fixed-point value of J where $C_L/\sin \alpha$ coefficients begin

ICC(3) = fixed-point value of J where $C_{D,i}/C_L^2$ coefficients begin

ICC(4) = fixed-point value of J where $C_{D,0}$ coefficients begin

For this purpose, all values in the COEFN(J) array are called coefficients (i.e., the t 's and the N_M 's are coefficients). The sequence of the sets is arbitrary, since changing the sequence requires only a change in the ICC(I) array. (See appendix I for Example II, the lunar orbiting probe.)

(17) The size of the integration steps is determined primarily by the error control variables. These are loaded as:

EREF = error reference value; δ in appendix D

ERLIMT = maximum value of δ that is acceptable on any particular step

EREF is always treated as a positive number; however, if it is loaded with a minus sign, this will cause error information to be printed at the completion of the problem. If no error control data is loaded, SUBROUTINE STDATA will set EREF = 1×10^{-6} , ERLIMT = 3×10^{-6} .

(18) The output control offers a choice on the frequency of output data as follows:

If MODOUT = 1, output will occur every n^{th} step ($n = \text{STEPS}$) until $t = \text{TMIN}$, and then MODOUT is set equal to 2 by the program

If MODOUT = 2, output occurs at equal time intervals of DELMAX until $t = \text{TMAX}$

If MODOUT = 3, output occurs at equal time intervals of DELMAX until $t = \text{TMIN}$, then MODOUT is set equal to 4 by the program

If MODOUT = 4, output occurs every n^{th} step ($n = \text{STEPS}$) until $t = \text{TMAX}$

STPEMX = maximum step limit before problem is completed

DELMAX = time interval between outputs

STEPS = number of steps between outputs

TMIN = time when MODOUT changes

Note that output control may, at times, strongly influence the integration step size especially if MODOUT is 2 or 3 and DELMAX is small. STDATA will put MODOUT = 4 and STEPS = 1.

Note that TMAX = time at start of a stage, plus the stage time, TB(NSTAGE), and is computed internally.

(19) Provision has been made to interrupt the integration procedure when an arbitrary value of an arbitrary parameter is attained. By interrupt it is meant that an output will occur at this point, input is permissible, and a decision is made whether to continue the stage, terminate the stage, or terminate the flight. Skip to instruction (20) if this facility is not desired. To cause an interrupt, set

LOOKX = COMMON C location of arbitrary parameter

XLOOK = value of C(LOOKX) where an interrupt is desired

INLOOK = input identification number for interrupt

END = a negative number if flight should be terminated, zero if stage should continue, or a positive number if stage should be terminated

If the interrupt is not desired the first time $C(LOOKX) = XLOOK$, set

LOOKSW = COMMON C location of a second arbitrary parameter

SWLOOK = value of $C(LOOKSW)$, which must be equaled or exceeded before an interrupt may occur (interrupt occurs if $C(LOOKX) = XLOOK$ and $C(LOOKSW) \geq SWLOOK$)

Typically, time may be the second arbitrary parameter; thus, STDATA sets $LOOKSW = 711$, the COMMON location of time. INLOOK of a nonzero value will cause any data cards of that identification number to be read-in prior to the interrogation of END. The order of the cards is discussed in instruction (24).

(20) Provision has been made to save a block of initial conditions and program control parameters prior to the integration of the n^{th} stage. This allows the flight to be flown again from the n^{th} stage onward with prescribed alterations. Skip to instruction (21) if this facility is not desired. To save the program control variable array, A, and the integration variable array, $XPRIM + XPRIMB$, just prior to integration of the n^{th} stage, set

NSAVE = the number of the n^{th} stage

The saved data, stored in the D array, will be returned to the A and $XPRIM + XPRIMB$ arrays after the flight is completed if

RECALL = any nonzero number

It is intended that changes in the succeeding flight will be made at the main input station ($\$DATA=1$). NSAVE and RECALL are not contained in the array A and are therefore unaffected by the save-recall sequence. The correct sequence of these controls is not always simple and an understanding of the main program and input stationing is quite desirable.

(21) If the standard set of data contained in the SUBROUTINE STDATA is not desired, set

CLEAR = any nonzero number

It is intended that this control shall be set nonzero by the $\$DATA = 99$ input station at the beginning of the main program. It is not affected by the save-recall sequencing explained in instruction (20).

(22) If the number of stages to be flown is not equal to the number of consecutive nonzero flight times, TB, set

LSTAGE = number of last stage to be flown

(23) When a transfer of origin occurs, provision has been made to read input into the program. This is done with the aid of \$DATA = 101, followed by the data statements desired.

(24) The sequencing of the input cards is not always simple and no rigid rules may be written down. Inspection of the program may be necessary to answer some questions. However, in general, the first input cards belong to the \$DATA = 300 group if an ephemerides tape is required. This group is followed by the \$DATA = 1 group, which consists of the main input for a single flight. Following this are the in-flight input cards, if any, which may be any combination of \$DATA = 101, \$DATA = INLOOK, or \$DATA = IDENT (NSTAGE) groups. The order of these groups of cards matches the order of the time sequence of events in the flight itself. For multiple flights, sets of the above groups may be added in tandem. It is usually desirable in this case, however, to read all the \$DATA = 300 sets at the same time (as in instruction (3)) to avoid excessive tape handling.

(25) Following is an input check list that may be helpful at execution time:

INPUT CHECK LIST^a

Takeoff time	Position and velocity (completely fill in one and only one block)			Reference and perturbing bodies	
	Rectangular	Orbit elements	Spherical	BODYCD =	
				Tape	Elliptic
DTOFFJ = TOFFT = TIME =	X = Y = Z = VX = VY = VZ = IMODE = 2	E = OMEGA = NODES = INCL = MA = P = IMODE = 1	LAT = LONG = AZI = ELEV = ALT = VEL = TKICK = IMODE = 4	bTAPE 3 = 0 bTBEGIN = bTEND = bELIST = TFILE =	ELIPS =
Output control	Error control	Restart feature	Parameter search	Atmosphere and coefficients	Oblateness - rotation
TMIN = MODOUT = STEPS = DELMAX = STEPMX =	EREF = ERLIMT =	NSAVE = RECALL = CLEAR =	LOOKX = XLOOK = LOOKSW = SWLOOK = INLOOK = END =	ATMN = RATM = COEFN = ICC = BETA =	Stage data OBLATN = ROTATE =
					TB = FLOW = SIMP = AREA = DELT = RMASS = AEXIT = IDENT = LSTAGE =

^aThe following standard data are loaded by SUBROUTINE STDATA:

DTOFFJ = 2440 000.0	MODOUT = 4	EREF = 1×10^{-6}
IMODE = 1	STEPS = 1.0	ERLIMT = 3×10^{-6}
BODYCD(1) = (ALF5)EARTH	STEPMX = 100.0	TFILE = 1.0
RMASS(1) = 1.0	LOOKSW = 711	

^bAt input 300, setting TAPE 3 = 0 is necessary to make an ephemeris tape.

APPENDIX H

PROGRAM LISTING

```

C      THIS MAIN PROGRAM IS THE SUPERSTRUCTURE ABOVE ALL SUBPROGRAMS.
C      SUBROUTINE TAPE CLEARS COMMON 1 THRU 4000 AND MAY CONSTRUCT AN
C      EPHEMERIS TAPE, ALSO, IT ALWAYS SETS TAPE3 =0. SUBROUTINE STDATA
C      LOADS A STANDARD SET OF DATA. IF RECALL DOES NOT EQUAL ZERO, A
C      PREVIOUSLY SAVED SET OF DATA(FROM STAGE) IS MOVED TO THE INITIAL
C      DATA LOCATION. THE MAIN INPUT STATION IS STATEMENT 8(INPUT1)
C      WHERE THE VEHICLE DATA FOR ALL STAGES MAY BE LOADED. SUBROUTINE ORDER IS
C      CALLED TO ORDER THE LIST OF BODIES, DETERMINE THE GRAVITATIONAL CONSTANT,
C      ORIGIN ROTATION RATE, ATMOSPHERIC RADIUS, RELOCATE ELLIPTIC EPHEMERIS DATA
C      AND POSITION THE EPHEMERIS TAPE.
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000),
C      1 TB(10), D(1100)
C
C      EQUIVALENCE
C      1(A ,C ( 11)),(B ,C (111)),(CLEAR ,C ( 3)),
C      2(D ,C (211)),(LSTAGE,A ( 38)),(NCASE ,C ( 1)),
C      3(NCASES,A ( 600)),(NSTAGE,A ( 3)),(RECALL,C ( 5)),
C      4(TB ,A ( 63)),(TAPE3 ,C ( 2)),(TABLE ,C (1911))
C
C      1 CALL INPUT (99,C,TABLE)
C      IF (TAPE3) 3,2,3
C      2 CALL TAPE
C      3 NCASE = NCASE + 1
C      WRITE OUTPUT TAPE 6,12,NCASE
C      12 FORMAT(12H1CASE NUMBER13,1H.)
C      IF (CLEAR) 5,4,5
C      4 CALL STDATA
C      5 IF (RECALL) 6,8,6
C      6 DO 7 J=1,1100
C      7 A(J) = D(J)
C      WRITE OUTPUT TAPE 6,16,NSTAGE,NCASES
C      16 FORMAT(13H RECALLED INITIAL DATA FROM STAGE12,8H OF CASE14,1H.)
C      8 CALL INPUT (1,C,TABLE)
C      IF (SENSE SWITCH 6) 13,14
C      13 WRITE OUTPUT TAPE 6,15
C      15 FORMAT(19H0EXIT VIA SENSE SW6)
C      CALL EXIT
C      14 IF (LSTAGE) 11,9,11
C      9 DO 10 LSTAGE=1,10
C      IF (TB(LSTAGE+1)) 10,11,10
C      10 CONTINUE
C      LSTAGE = 10
C      11 CALL ORDER
C      17 CALL STAGE
C      GO TO 1
C      END

```

SUBROUTINE AERO

```

C      SUBROUTINE AERO COMPUTES THE LIFT AND DRAG ACCELERATIONS. AS IN SUBROUT-
C      INE THRUST, THESE VECTORS ARE REFERENCED TO THE RELATIVE WIND VELOCITY.
C      COEFFICIENTS OF LIFT, INDUCED DRAG, AND DRAG AT ZERO ANGLE OF ATTACK ARE
C      ASSUMED TO BE FUNCTIONS OF MACH NUMBER AND ANGLE OF ATTACK. TABLES OF
C      CDI/CL**2, CL/SIN(ALPHA), AND CDO ARE ASSUMED AS FITTED QUADRATIC EQUAT-
C      IONS IN THE COEFN ARRAY. GASFAC IS THE SQRT(SPECIFIC HEAT RATIO * STAND-
C      ARD ACCELERATION OF GRAVITY * UNIVERSAL GAS CONSTANT). FOR EARTH, GASFAC=
C      20.064881 (METERS / SEC**2 / KELVIN DEGREE)**1/2.
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000),
C      1 VATM(3), P(3), XIFT(3), DRAG(3), PAR(3), X(100)
C
C      EQUIVALENCE
C      1(A ,C ( 11)),(ALPHA ,A ( 49)),(AREA ,B ( 61)),
C      2(B ,C (1111)),(BETA ,A ( 50)),(CD ,A ( 165)),
C      3(CDI ,A ( 163)),(CL ,A ( 164)),(COSALF,B ( 48)),
C      4(COSBET,B ( 49)),(DNSITY,B ( 29)),(DRAG ,B ( 69)),
C      5(GASFAC,A ( 46)),(P ,B ( 84)),(PAR ,B ( 60)),
C      6(PMAGN ,B ( 50)),(IQ ,B ( 59)),(R ,B ( 102)),
C      7(SINALF,B ( 46)),(SINBET,B ( 47)),(TM ,B ( 34)),
C      8(VATM ,B ( 97)),(VMACH ,B ( 38)),(VQ ,B ( 100)),
C      9(VQSQRD,B ( 101)),(X ,B ( 401)),(XIFT ,B ( 72))
C
C      Q = 0.5*DNSITY*VQSQRD
C      QVAL = Q*AREA/X(2)
C      VMACH=SQRTF(VQSQRD/TM)/GASFAC
C
C      COMPUTE THE X,Y,Z COMPONENTS OF LIFT.
C      IF (ALPHA) 2,1,2
C      1 CL = 0.0
C      CDI=0.0
C      GO TO 4
C      2 CL = QUAD(VMACH,2)*SINALF
C      AA = QVAL*CL/PMAGN
C      AB = SINBET/VQ
C      DO 3 K=1,3
C      3 XIFT(K) = AA*(AB*PAR(K)+COSBET*P(K))
C      7 CDI=QUAD(VMACH,3)*CL*CL
C
C      COMPUTE THE X,Y,Z COMPONENTS OF DRAG.
C      4 CD = CDI*QUAD(VMACH,4)
C      AC = -CD*QVAL/VQ
C      DO 5 K=1,3
C      5 DRAG(K) = AC*VATM(K)
C      6 RETURN
C      END

```

```

C               FUNCTION ARCTAN (Y,X)
C   THE FORTRAN II LIBRARY ATANF(+ OR - Z=TAN(THETA)) USES A SINGLE
C   ARGUMENT WITH ITS SIGN TO GIVE THETA IN THE FIRST (+2) OR FOURTH
C   (-2) QUADRANT.
C
C   THE ARCTAN FUNCTION MAY BE USED IF + OR - Z IS DERIVED FROM A
C   FRACTION SO THAT ARCTAN (Y,X) = TAN-1 ((+OR-Y=SIN(THETA))/(+OR-X=
C   COS(THETA))). THUS THE ARCTAN (Y,X) GIVES THETA IN ITS PROPER
C   QUADRANT FROM -180 DEGREES TO +180 DEGREES.
C
C   IF (X) 2,1,2
1  ARCTAN=SIGNF(1.57079632,Y)
  GO TO 4
2  ARCTAN=ATANF(Y/X)
  IF(X) 3,1,4
3  ARCTAN=ARCTAN+SIGNF(3.14159265,Y)
4  RETURN
  END

```

```

C               SUBROUTINE CONV1 (V,AMC)
C   THIS ROUTINE COMPUTES -- (1) ANGULAR MOMENTUM, AMC(4)
C                           (2) ANGULAR MOMENTUM SQUARED, AMC(5)
C                           (3) X,Y,Z COMPONENTS OF ANG. MOM., AMC(J)
C                           (4) VELOCITY, V(4)
C                           (5) VELOCITY SQUARED, V(5)
C
C   COMMON C
C
C   DIMENSION A(600), B(700), C(4000),
1  AMC(3), V(5), RB(3), IND(3)
C
C   EQUIVALENCE
1(A ,C ( 11)),(B ,C (111)),(IND ,A ( 60)),
2(RB ,B ( 193))
C
C   DO 1 J1=1,3
  J2=IND(J1)
  J3=IND(J2)
1  AMC(J3) = Rb(J1)*V(J2)-RB(J2)*V(J1)
  AMC(5) = AMC(1)**2+AMC(2)**2+AMC(3)**2
  AMC(4) = SQRTF(AMC(5))
  V(5) = V(1)**2+V(2)**2+V(3)**2
  V(4) = SQRTF(V(5))
  RETURN
  END

```

```

C               SUBROUTINE CONV2
C   THIS ROUTINE CONVERTS RECTANGULAR COORDINATES INTO ORBIT ELEMENTS.
C   RECTANGULAR COORDINATES- POSITION COMPONENTS,X,AND VELOCITY COMPONENTS,VX.
C   THE ORBIT ELEMENTS ARE IN THE ORBELS ARRAY-
C   (1) ECCENTRICITY (4) INCLINATION
C   (2) ARGUMENT OF PERICENTER (5) MEAN ANOMALY
C   (3) LONGITUDE OF ASCENDING NODE (6) SEMILATUS RECTUM
C
C   COMMON C
C
C   DIMENSION A(600), B(700), C(4000),
1  AMC(3), ORBELS(6), RB(3)
C
C   EQUIVALENCE
1(A ,C ( 11)),(AM ,B ( 90)),(AMSQRD,B ( 91)),
2(AMC ,B ( 87)),(B ,C (111)),(COSTRU,B ( 53)),
3(EPAR ,B ( 26)),(GK2M ,B ( 36)),(ORBELS,B ( 116)),
4(R ,B ( 102)),(RB ,B ( 193)),(SINTRU,B ( 52)),
5(STRU ,B ( 40)),(V ,B ( 95)),(VSQRD ,B ( 96)),
6(VX ,B ( 92)),(VY ,B ( 93)),(VZ ,B ( 94))
C
C   ORBELS(6)=AMSQRD/GK2M
  R=SQRTF(RB(1)**2+RB(2)**2+RB(3)**2)
  TRU=ARCTAN(AM/GK2M*(RB(1)*VX+RB(2)*VY+RB(3)*VZ),ORBELS(6)-R)
  IF(AMC(1)) 2,1,2
1  ORBELS(3)=0.
  GO TO 3
2  ORBELS(3)=ARCTAN(AMC(1),-AMC(2))
3  ORBELS(4)=ARCTAN(SQRTF(AMC(1)**2+AMC(2)**2),AMC(3))
  SNODE=SINF(ORBELS(3))
  CNODE=COSE(ORBELS(3))
  AA=RB(1)*CNODE+RB(2)*SNODE
  AB=RB(3)*SINF(ORBELS(4))+COSE(ORBELS(4))*(RB(2)*CNODE-RB(1)*SNODE)
  ORBELS(2)=ARCTAN(AB,AA)-TRU
  ORBELS(1)=SQRTF(ABSF(1.+ORBELS(6)*(VSQRD/GK2M-2./R)))
  EPONE=SQRTF(1.+ORBELS(1))
  E2M1=1.-ORBELS(1)**2
  EPAR=SQRTF(ABSF(E2M1))
  SINTRU=SINF(STRU)
  COSTRU=COSE(STRU)
  EPAS=SQRTF(ABSF(1.-ORBELS(1)))*SINTRU/(1.+COSTRU)
  ETHETA=ORBELS(1)*SINTRU/(1.+ORBELS(1)*COSTRU)*EPAR
4  IF(E2M1) 5,6,6
5  ORBELS(5)=LOGF((EPONE+EPAS)/(EPONE-EPAS))-ETHETA
  GO TO 7
6  ORBELS(5)=2.*ARCTAN(EPAS,EPONE)-ETHETA
7  RETURN
  END

```

```

C                                     SUBROUTINE ERRORZ
C THIS SUBROUTINE COMPUTES THE RELATIVE ERRORS BETWEEN THE R-K AND LOW-ORDER
C INTEGRATION SCHEMES. IT ALSO COMPUTES THE ERROR COEFFICIENT, A, AND SAVES
C THE ERROR DATA WHEN EREF HAS A - SIGN. THE BRANCH ON IMODE DETERMINES
C WHICH SET OF NORMALIZING FACTORS ARE TO BE USED.
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000),
C      1 RELERR (8), XPRIM (200), XINC (100)
C
C      EQUIVALENCE
C      1(A ,C ( 11)),(A1 ,B ( 10)),(A2 ,B ( 11)),
C      2(B ,C (111)),(DELT ,B ( 1)),(E2 ,B ( 18)),
C      3(EREF ,A ( 13)),(IMODE ,A ( 1)),(INDERR,B ( 51)),
C      4(R ,B ( 102)),(STEPGO,A ( 41)),(STEPNO,A ( 42)),
C      5(V ,B ( 95)),(XINC ,B ( 601)),(XPRIM ,C ( 711))
C      EQUIVALENCE (RELERR,XINC)
C
C      E2 = 0.
C      RELERR(2) = XINC(2)/XPRIM(2)
C      IF (IMODE-1) 2,1,2
C
C      COMPUTE THE NORMALIZED INTEGRATION ERRORS FOR THE ORBIT ELEMENTS.
C      1 RELERR(3)=XINC(3)/(XPRIM(3)+1.)/10.
C      RELERR(8)=XINC(8)/XPRIM(8)/10.
C      DD 10 J=1,4
C      10 RELERR(J+3)=XINC(J+3)/62.831853
C      GO TO 3
C
C      COMPUTE THE NORMALIZED INTEGRATION ERRORS IN RECTANGULAR VARIABLES.
C      2 V1 = V+100.
C      DD 20 J=1,3
C      RELERR(J+2)=XINC(J+2)/V1
C      20 RELERR(J+5)=XINC(J+5)/R
C
C      SELECT MAXIMUM ERROR, COMPUTE ERROR COEFFICIENT, POSSIBLY SAVE ERROR DATA.
C      3 DO 5 J=2,8
C      IF (ABSF(RELERR(J))-E2) 5,5,4
C      4 K=J
C      E2 = ABSF(RELERR(J))
C      5 CONTINUE
C      E2 = E2 + 2E-8
C      A1 = A2
C      A2 = LOGF(E2)-5.*LOGF(ABSF(DELT))
C      IF (EREF) 6,7,7
C      6 WRITE TAPE 4,STEPGO,STEPNO,XPRIM(1),DELT,A2,E2,(RELERR(J),J=2,8),K
C      INDERR = INDERR + 1
C      7 RETURN
C      END

```

SUBROUTINE EQUATE

```

C THIS SUBROUTINE IS CALLED FROM NBODY TO EVALUATE THE DERIVATIVES OF THE
C VARIABLES OF INTEGRATION. EITHER RECTANGULAR COORDINATES OR ORBIT ELE-
C MENTS MAY BE USED AS THE VARIABLES OF INTEGRATION, BUT IN THE CASE OF THE
C LATTER, THE CORRESPONDING RECTANGULAR COORDINATES MUST FIRST BE FOUND.
C THIS IS DONE AT THE BEGINNING THRU THE USE OF KEPLERS EQUATION. THE
C PERTURBATING ACCELERATIONS ARE FOUND BY CALLING VARIOUS OTHER SUBROUTINES
C AND THEIR SUM RESOLVED ALONG THE X,Y,Z AXIS. FINALLY, THE DERIVATIVES
C ARE CALCULATED. IN THE CASE OF ORBIT ELEMENTS, THE X,Y,Z PERTURBATING
C ACCELERATION COMPONENTS MUST FIRST BE RESOLVED INTO CIRCUMFERENTIAL,RADIAL
C AND NORMAL COMPONENTS. THIS ROUTINE ALSO CHANGES THE INTEGRATION VARI-
C ABLES FROM ORBIT ELEMENTS TO RECTANGULAR VARIABLES IF THE ECCENTRICITY
C APPROACHES UNITY. THE X,XPRIM, AND XDOT ARRAYS ARE AS FOLLOWS.
C
C      X          ORBIT ELEMENTS          RECTANGULAR COORDINATES
C
C      1          TIME                     TIME
C      2          MASS                     MASS
C      3          ECCENTRICITY             X-VELOCITY
C      4          ARGUMENT OF PERICENTER   Y-VELOCITY
C      5          ARGUMENT OF ASC. NODE    Z-VELOCITY
C      6          INCLINATION              X
C      7          MEAN ANOMALY             Y
C      8          SEMILATUS RECTUM         Z
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000),
C      1 XPRIM (100,2), VX (3), QX (3),
C      2 RB (3), NEFMRS (8), X (100),
C      3 XPRIMB (100,2), FORCE (3), XIFT (3),
C      4 DRAG (3), OBLAT (3), COMPA (3),
C      5 XDOT (100)

```

```

EQUIVALENCE
1(A,C ( 11)),(AMSQRD,B ( 91)),(ASYMPT,A ( 7)),
2(B,C (111)),(BNAME,B ( 122)),(CINCL,B ( 55)),
3(CIRCUM,B ( 82)),(COMPA,B ( 63)),(CONSTU,A ( 32)),
4(COSTRU,B ( 53)),(COSV,B ( 57)),(DRAG,B ( 69)),
5(EMUNE,B ( 28)),(EPAR,B ( 26)),(ETUL,A ( 30)),
6(EXMODE,B ( 27)),(FLOW,B ( 51)),(FORCE,B ( 66)),
7(GK2M,B ( 36)),(GKM,B ( 37)),(IMUDE,A ( 11)),
8(KSUB,B ( 19)),(MBODYS,B ( 42)),(NEFMRS,B ( 185)),
9(INSTART,B ( 24)),(OBLATN,A ( 40)),(OBLAT,B ( 75))
EQUIVALENCE
1(PRESS,B ( 33)),(PUSHO,B ( 39)),(QX,B ( 78)),
2(RADIAL,B ( 81)),(RATMUS,B ( 23)),(RB,B ( 193)),
3(R,B ( 102)),(RSQRD,B ( 45)),(SINCL,B ( 54)),
4(SINTRU,B ( 52)),(SINV,B ( 56)),(SPU,A ( 44)),
5(TABL,B ( 20)),(TOFFT,A ( 24)),(TRANSFER,B ( 8)),
6(TTEST,A ( 54)),(U,A ( 59)),(V,B ( 95)),
7(VSQRD,B ( 96)),(VX,B ( 92)),(XDOT,B ( 50)),
8(XIFT,B ( 72)),(XPRIM,C ( 71)),(XPRIMB,C ( 91)),
9(X,B ( 40)),(ZORMAL,B ( 83)),(ZN,B ( 43))
C
      TABLT=X(1)/SPU+TOFFT
      IMODE=IMODE
1 GO TO (2,16,16),IMODE
C
C STATEMENTS 2 TO 16 FIND THE RECTANGULAR POSITION AND VELOCITY FROM ORBIT
C ELEMENTS AND TRUE ANOMALY. THE TRUE ANOMALY IS FOUND FROM ITERATIVE
C SOLUTION OF KEPLERS EQUATION.
2 E2 = X(3)**2
  E2M1 = 1.-E2
  EMONE = X(3)-1.
  EPAR=SQRTF(ABSF(E2M1))
  VCIRCL=GKM/SQRTF(X(8))
C
C COMPUTE SINE AND COSINE OF TRUE ANOMALY.
C PART A. E=1
3 IF (EMONE) 10,4,5
4 SINTRU = 0.
  COSTRU = 1.
  GO TO 14
C
C PART d. E IS GREATER THAN 1
5 DO 7 J=1,100
  DELM = X(7)-U+X(3)*SINH(U)
  ECOSU = X(3)*COSH(U)
  DELU = DELM/(1.0-ECOSU)
  U = U+DELU
6 IF (ABSF(DELM)-CONSTU) 9,9,7
7 CONTINUE
  ASYMPT = 1.0
  IF (MBODYS) 8,23,8
8 CALL EPHMRS
  GO TO 23
9 COSU = COSH(U)
  DEM1 = 1.-X(3)*COSU
  COSTRU = (COSU-X(3))/DEM1
  SINTRU = -EPAR*SINH(U)/DEM1
  GO TO 14
C
C PART C. E IS LESS THAN 1
10 DO 12 J=1,15
  DELM = X(7)-U+X(3)*SINF(U)
  ECOSU = X(3)*COSF(U)
  DELU = DELM/(1.0-ECOSU+0.01*ECOSU**3)
  U = U+DELU
11 IF (ABSF(DELM)-CONSTU) 13,13,12
12 CONTINUE
  WRITE OUTPUT TAPE 6,55,U,DELU
  CALL EXIT
13 COSU = COSF(U)
  DEM1 = 1.-X(3)*COSU
  COSTRU = (COSU-X(3))/DEM1
  SINTRU = EPAR*SINF(U)/DEM1
14 PDVR = 1.+X(3)*COSTRU
C
C COMPUTE POSITION AND VELOCITY FROM ORBIT ELEMENTS AND TRUE ANOMALY.
C ALSO, CLEAR THE PERTURBATING ACCELERATIONS.
15 SOMEGA = SINF(X(4))
  COMEGA = COSF(X(4))
  SNODE = SINF(X(5))
  CNODE = COSF(X(5))
  SINCL = SINF(X(6))
  CINCL = COSF(X(6))
  SINV=SINTRU*COMEGA+COSTRU*SOMEGA
  COSV=COSTRU*COMEGA-SINTRU*SOMEGA
  AR=COSV*CNODE-SINV*SNODE*CINCL
  B1=SINV*CNODE+COSV*SNODE*CINCL
  C1=COSV*SNODE+SINV*CNODE*CINCL
  D1=SINV*SNODE-COSV*CNODE*CINCL
  E1 = X(3)*SOMEGA+SINV
  F1 = X(3)*COMEGA+COSV
  AS=E1*CNODE+F1*SNODE*CINCL
  B2=F1*CNODE-CINCL-E1*SNODE
  R = X(8)/PDVR
  RSQRD = R*R
  SINVV=SINV*SINCL
  RB(1) = R*AR
  RB(2) = R*C1
  RB(3) = R*SINNV
  VX(1)=-VCIRCL*AS
  VX(2)=VCIRCL*B2
  VX(3)=VCIRCL*F1*SINCL
  GO TO 18

```

```

C
16 DO 17 K=1,3
   VX(K) = X(K+2)
17 RB(K) = X(K+5)
   RSQRD = RB(1)*RB(1) + RB(2)*RB(2) + RB(3)*RB(3)
   R=SQRT(RSQRD)
18 VSQRD=VX(1)*VX(1)+VX(2)*VX(2)+VX(3)*VX(3)
   V = SQRT(VSQRD)
   DO 19 I=1,15
19 FORCE(I) = 0.
C
C   TEST FOR PRESENCE OF PERTURBING BODIES.
C   IF (MBODYS) 20,21,20
20 CALL EPHMRS
21 IF (XABSF(IMODE)-1) 26,22,26
C
C   TEST FOR CHANGE FROM ORBIT ELEMENTS TO TEMPORARY RECTANGULAR
C   COORDINATES IF E IS TOO NEAR TO UNITY.
22 IF (ETOL-ABS(FEMONE)) 26,23,23
23 IF (IMODE) 54,24,24
24 IMODE=-3
   IF (NSTART) 25,54,25
25 TTEST = X(1)
27 CALL TESTTR
C
C   TEST FOR OBLATENESS PERTURBATION COMPUTATION.
26 IF(OBLATN-BNAME)30,29,30
29 CALL OBLATE
C
C   TEST FOR PRESENCE OF THRUST.
30 XDOT(2) = -FLOW
   IF (R-RATMOS) 31,31,32
31 CALL ICAO
   GO TO 33
32 PRESS=0.
33 IF (PUSH0) 37,36,37
36 ASSIGN 40 TO NDONE
   GO TO 38
37 CALL THRUST
   ASSIGN 41 TO NDONE
C
C   TEST FOR EXISTENCE OF ATMOSPHERE.  FIND AERODYNAMIC FORCES.
38 IF (PRESS ) 39,42,39
39 GO TO NDONE, (40,41)
40 CALL THRUST
41 CALL AERO
C
C   SUM COMPONENTS OF THE PERTURBING ACCELERATION.
42 DO 43 J=1,3
43 COMPA(J) = -QX(J)+OBLAT(J)+FORCE(J)+XIFT(J)+DRAG(J)
44 GO TO (47,45,45),IMODE
C
C   COMPUTE DERIVATIVES FOR THE RECTANGULAR VARIABLES OF INTEGRATION.
45 AA = GK2M/R/RSQRD
   DO 46 K=1,3
   XDOT(K+5) = X(K+2)
46 XDOT(K+2) = COMPA(K)-AA*X(K+5)
   GO TO 54
C
C   COMPUTE THE DERIVATIVES OF THE ORBIT ELEMENTS.  (AFTER RESOLVING
C   PERTURBATING ACCELERATION INTO CIRCUMFERENTIAL, RADIAL, NORMAL COMPONENTS)
47 CIRCUM=COMPA(3)*COSV*SINCL-COMPA(1)*B1-COMPA(2)*D1
   RADIAL=COMPA(1)*AR+COMPA(2)*C1+COMPA(3)*SINVY
   ZORMAL=COMPA(1)*SNODE*SINCL-COMPA(2)*CNODE*SINCL+COMPA(3)*CINCL
   ZN=VCIRCL*E2M1*EPAR/X(8)
   RDVPP1 = 1./PDVR + 1.
   RDVA = E2M1/PDVR
   XDOT(8) = 2.*R/VCIRCL*CIRCUM
   IF (X(3)) 48,48,49
48 CSQRD = CIRCUM*CIRCUM
   RASQRD = RADIAL*RADIAL
   DEM1 = (4.*CSQRD+RASQRD)*VCIRCL
C
C   TEST FOR IN-PLANE PERTURBATION.
   IF (DEM1) 57,56,57
56 XDOT(3) = 0.
   XDOT(4) = 0.
   XDOT(7) = 0.
   GO TO 50
57 VDV2R=VCIRCL/R/2.
   XDOT(3) = SQRT(4.*CSQRD+RASQRD)/VCIRCL
   XDOT(4) = VDV2R*(2.*CSQRD+RASQRD)/DEM1*RADIAL
   XDOT(7) = ZN-VDV2R*(6.*CSQRD+RASQRD)/DEM1*RADIAL
   GO TO 50
49 XDOT(3) = (SINTRU*RADIAL+(PDVR-ROVA)/X(3)*CIRCUM)/VCIRCL
   XDOT(4) = (SINTRU/X(3)*RDVPP1+CIRCUM-COSTRU*RADIAL/X(3))/VCIRCL
   XDOT(7) = ZN+EPAR/VCIRCL*((COSTRU/X(3)-2./PDVR)*RADIAL-(SINTRU/X(3)*
   IROVPP1*CIRCUM))
50 IF(SINCL) 51,52,51
51 XDOT(5) = SINV/SINCL*ZORMAL/VCIRCL/PDVR
   GO TO 53
52 XDOT(5) = 0.
53 XDOT(6) = COSV*ZORMAL/PDVR/VCIRCL
54 RETURN
55 FORMAT(41HOKEPLERS EQUATION CONVERGENCE FAILURE, U=G15.8,7H DELU=
   I615.8)
   END

```

```

C SUBROUTINE EPHMRS
C SUBROUTINE EPHMRS IS CALLED TO COMPUTE THE POSITIONS OF THE PERTURBING
C BODIES RELATIVE TO THE VEHICLE AND, FROM THESE, THEIR PERTURBING ACCELE-
C TIONS UPON THE VEHICLE. OCCASIONALLY THIS ROUTINE IS CALLED FOR THE PURPOSE
C OF TRANSLATING THE ORIGIN IN WHICH CASE (TRSFER=1) THE RELATIVE VELOCITIES
C ARE ALSO CALCULATED. IF A BODYS POSITION IS TO BE COMPUTED FROM AN ELLIPTIC
C APPROXIMATION SUBROUTINE ELIPSE IS CALLED. OTHERWISE, THE POSITION WILL BE
C CALCULATED IN EPHMRS FROM THE PRECISION TAPE EPHEMERIS. THE DO 19 LOOP
C ENCOMPASSES ALMOST THE ENTIRE EPHMRS SUBROUTINE AND, IN EFFECT, ELIPSE TOO.
C
C COMMON C
C
C DIMENSION A(600), B(700), C(4000),
C 1 QX(3), IBODY(8), EFMRS(7), XP(3,8), RB(3,8), R (8), TIM(7),
C 2 NEFMRS(8), TDATA(6,3,7), TDEL(7), BMASS(8), VEFM(3,8), DATA(21)
C 3 , TDATA(18,7)
C
C EQUIVALENCE
C 1(A ,C ( 11)),(AU ,A ( 29)),(B ,C (111)),
C 2(BMASS ,B ( 137)),(DTOFFJ,A ( 23)),(EFMRS ,B ( 130)),
C 3(IBODY ,B ( 177)),(MBODY5,B ( 42)),(NEFMRS,B ( 185)),
C 4(QX ,B ( 78)),(R ,B ( 102)),(RB ,B ( 193)),
C 5(SQRDK ,B ( 35)),(SPD ,A ( 44)),(TABLT ,B ( 20)),
C 6(TDATA ,B ( 265)),(TDEL ,B ( 170)),(TIM ,B ( 163)),
C 7(TRSFER,B ( 8)),(VEFM ,B ( 241)),(XP ,B ( 217))
C EQUIVALENCE (IBF,FIB), (TDAT,TDATA)
C
C PART 2. SET INDEXS, FIND POSITION IF ELLIPSE IS USED (NEFMRS = 20 OR UP).
C DO 19 JB=1,MBODYS
C JB1 = JB+1
C IBF = IBODY(JB1)
C IB = XABSF(IBF)
C IF (NEFMRS(JB)-20) 2,2,1
C 1 CALL ELIPSE (JB1)
C IF (TRSFER) 12,12,17
C
C PART 3. TAPE EPHEMERIS IS TO BE USED. FIND DIFFERENCE (DT) BETWEEN
C CURRENT PROBLEM TIME (DTOFFJ+TABLT) AND MIDPOINT TIME (TIM) OF CURRENTLY
C STORED TAPE DATA. THEN SEE IF CURRENT DATA IS OKAY. TDEL = TIME INTERVAL
C ON EITHER SIDE OF TIM FOR WHICH CURRENT DATA IS GOOD.
C 2 DT = TABLT - (TIM(JB) - DTOFFJ)
C IF (ABSF(DT)-TDEL(JB)) 10,10,3
C
C PART 4A. CURRENT DATA NOT OKAY. READ IN NEXT DATA SET. IF DT IS -,
C BACK UP THE TAPE 2 RECORDS BEFORE READING.
C 3 IF (DT) 4,5,5
C 4 BACKSPACE 3
C BACKSPACE 3
C 5 READ TAPE 3, (DATA(1), I=1,21)
C
C PART 4B. IF THIS DATA IS FOR A BODY IN THE BNAME LIST, STOKE IT.
C (IF NOT STOKED, WE MIGHT HAVE TO RETURN FOR IT.) IF ELLIPSE DATA IS
C PROVIDED FOR THE BODY FOUND, BY-PASS THE TAPE DATA AND READ IN NEXT SET.
C DO 7 J = 1,MBODYS
C IF ((DATA(1)+EFMRS(J))*(-(DATA(1)+EFMRS(J)))) 7,6,7
C 6 IF (NEFMRS(J)-20) 8,8,3
C 7 CONTINUE
C GO TO 3
C
C PART 4C. MOVE THE DATA INTO PLACE AND THEN GO BACK AND SEE IF IT IS OKAY.
C 8 TIM(J) = DATA(2)
C TDEL(J) = DATA(3)
C DO 9 JJ=1,18
C TDATA(JJ,J) = DATA(JJ+3)
C 9 CONTINUE
C GO TO 2
C
C PART 5. CURRENT DATA IS OKAY. GET POSITION FROM THE POLONOMIAL
C P = A + BX + CX**2 + DX**3 + EX**4 + FX**5.
C 10 DO 11 K=1,3
C XP(K,JB1) = TDATA(1,K,JB)
C DO 11 KT=2,6
C XP(K,JB1) = XP(K,JB1)* DT +TDATA(KT,K,JB1)
C 11 CONTINUE
C IF (TRSFER) 12,12,15
C
C PART 6. COMPUTE DISTANCE FROM REFERENCE AND FROM ROCKET .
C 12 DO 13 K=1,3
C XP(K,JB1) = XP(K,IB) +XP(K,JB1)*SIGNF(AU,FIB)
C 13 RB(K,JB1)= RB(K,1) - XP(K,JB1)
C
C PART 7. COMPUTE PERTURBING ACCELERATIONS (QX). 4194304=2**22 IS REMOVED
C TO PREVENT OVERFLOW. 2048=2**11 AND 8589934592=2**33 RESTORE THE SCALE.
C PRSQRD = (RB(1,JB1)**2 + RB(2,JB1)**2 + RB(3,JB1)**2)/4194304.
C RRELL = SQRTF(PRSQRD)
C RSQRD = ( XP(1,JB1)**2 + XP(2,JB1)**2 + XP(3,JB1)**2)/4194304.
C RCUBE = RSQRD * SQRTF(R SQRD)
C PRCUBE = PRSQRD * RRELL
C R(JB1) = RRELL*2048.
C DO 14 K=1,3
C 14 QX(K)=SQRD * BMASS(JB1) * ((XP(K,JB1)/RCUBE) + RB(K,JB1)/PRCUBE)/
C 1 8589934592. + QX(K)
C GO TO 19
C
C PART 8. COMPUTE VELOCITY FROM V = B + 2CX + 3DX**2 + 4EX**3 + 5FX**4
C AND FROM REFERENCE BODY VELOCITY (VEFM(IB)).
C 15 DO 16 K=1,3
C VEFM(K,JB1) = 0.
C DO 16 KT=1,5
C 16 VEFM(K,JB1) = (VEFM(K,JB1)*DT+TDATA(KT,K,JB)*FLOATF(-KT*6))
C 17 DO 18 K=1,3
C 18 VEFM(K,JB1) = VEFM(K,IB) + VEFM(K,JB1)*SIGNF(AU/SPD,FIB)
C GO TO 12
C 19 CONTINUE
C RETURN
C END

```

```

C      SUBROUTINE EXTRA
C
C      THIS ROUTINE IS EXECUTED BETWEEN FLIGHTS AND MAY THEREFORE BE EXPANDED TO
C      DO ADDITIONAL COMPUTATION BETWEEN SUCCESSIVE FLIGHTS.
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000)
C
C      EQUIVALENCE
C      1(A ,C ( 11)), (B ,C (1111)), (QMAX ,B ( 44)),
C      2(SIGNAL,B ( 31))
C
C      SIGNAL = 0.
C      QMAX = 0.
C      RETURN
C      END

C      SUBROUTINE EXTRAS
C
C      THIS ROUTINE IS EXECUTED BETWEEN STAGES AND MAY THEREFORE BE EXPANDED TO
C      DO CALCULATIONS BETWEEN SUCCESSIVE STAGES OF A FLIGHT.
C
C      RETURN
C      END

C      SUBROUTINE ELIPSE (JB1)
C
C      THIS SUBROUTINE IS CALLED FROM EPHMRS TO COMPUTE THE POSITION OF A BODY
C      USING APPROXIMATE ELLIPTIC DATA. THE VELOCITY IS ALSO COMPUTED IF THE
C      ORIGIN IS BEING TRANSLATED (TRANSFER=1.0). THE ELLIPSE DATA IS READ FROM
C      INPUT CARDS AND ORGANIZED IN SUBROUTINE ORDER. TPD IS TIME SINCE PERIHELION
C      PASSAGE, ZM IS MEAN ANOMALY, U IS ECCENTRIC ANOMALY.
C
C      TDATA ARRAY - (K) SEMILATUS RECTUM (K+7) PERIOD
C                   (K+1) ECCENTRICITY (K+8) SIN OMEGA
C                   (K+2) OMEGA (K+9) SIN NODE
C                   (K+3) NODE (K+10) SIN INCLINATION
C                   (K+4) INCLINATION (K+11) COS OMEGA
C                   (K+5) JD OF PERIHELION (K+12) COS NODE
C                   (K+6) FRACTIONAL PART OF (K+5) (K+13) COS INCLINATION
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000),
C      1 XP(3,8), VEFM(3,8), TDATA(121)
C
C      EQUIVALENCE
C      1(A ,C ( 11)), (B ,C (1111)), (CONSU ,A ( 31)),
C      2(DTOFFJ,A ( 23)), (TABLT ,B ( 20)), (TDATA ,B ( 265)),
C      3(TRANSFER,B ( 8)), (VEFM ,B ( 241)), (XP ,B ( 217))
C
C      K = 18*(JB1-2)+1
C      TPD = (DTOFFJ-TDATA(K+5))+(TABLT-TDATA(K+6))
C      ZM = 6.28318533/TDATA(K+7)
C      ZM = ZM*MODF(TPD,TDATA(K+7))
C
C      GET THE SINE(SINTRU) AND THE COSINE (COSTRU) OF THE TRUE ANOMALY
C      BY ITERATING KEPLERS EQUATION. THEN COMPUTE X,Y,Z (XP).
C      U = ZM+TDATA(K+1)*SINF(ZM)+.5*TDATA(K+1)**2*SINF(2.*ZM)
C      DO 1 J=1,10
C      DELM = ZM-U+TDATA(K+1)*SINF(U)
C      DELU = DELM/(1.-TDATA(K+1)*COSF(U))
C      U = U+DELU
C      IF (ABSF(DELM)-CONSU) 2,2,1
C      1 CONTINUE
C      2 COSU = COSF(U)
C      DENOM = 1.-TDATA(K+1)*COSU
C      COSTRU = (COSU-TDATA(K+1))/DENOM
C      R = TDATA(K)/(1.+TDATA(K+1)*COSTRU)
C      SINTRU = SQRTF(1.-TDATA(K+1)**2)*SINF(U)/DENOM
C      SINV = SINTRU*TDATA(K+11)+COSTRU*TDATA(K+8)
C      COSV = COSTRU*TDATA(K+11)-SINTRU*TDATA(K+8)
C      XP(1,JB1) = R*(COSV*TDATA(K+12)-SINV*TDATA(K+9)+TDATA(K+13))
C      XP(2,JB1) = R*(COSV*TDATA(K+9)+SINV*TDATA(K+12)+TDATA(K+13))
C      XP(3,JB1) = R*SINV*TDATA(K+10)
C      IF (TRANSFER) 3,4,3
C
C      COMPUTE THE VELOCITIES FOR THE TRANSFER OF ORIGIN.
C      3 EX = TDATA(K+1)*TDATA(K+8)+SINV
C      FX = TDATA(K+1)*TDATA(K+11)+COSV
C      CFACT = ZM*TDATA(K)/(SQRTF(1.-TDATA(K+1)**2)**3)
C      AX = EX*TDATA(K+12)+FX*TDATA(K+9)+TDATA(K+13)
C      BX = FX*TDATA(K+12)+EX*TDATA(K+13)-EX*TDATA(K+9)
C      VEFM(1,JB1) = -AX*CFACT
C      VEFM(2,JB1) = BX*CFACT
C      VEFM(3,JB1) = FX*CFACT+TDATA(K+10)
C      4 RETURN
C      END

```

```

      REM      SUBROUTINE EXADD (A,B,C)
      REM THIS ROUTINE WILL ADD IN DOUBLE PRECISION A QUANTITY C TO THE DOUBLE
      REM PRECISION VARIABLE A+B WHERE A IS THE MOST SIGNIFICANT PART AND B IS
      REM THE LEAST SIGNIFICANT PART.
      ENTRY    EXADD
      COMMON   -206
Q1    COMMON  1
Q2    COMMON  1
TEMP1 COMMON  1
TEMP2 COMMON  1
      HTR
      BCI      1,EXADD
      SXD      4,1
      SXD      4,2
EXADD  SXD      4,4
      CLA*     1,4
      FAD*     3,4
      STQ      Q1
      FAD*     2,4
      STQ      Q2
      FAD      Q1
      STQ      Q1
      STO      TEMP1
      CLA      Q1
      FAD      Q2
      STO      TEMP2
      FAD      TEMP2
      FAD      TEMP1
      STQ      Q1
      FSB      TEMP2
      STO*     1,4
      STQ      Q2
      CLA      Q1
      FAD      Q2
      STO*     2,4
      TRA      4,4
      END

      SUBROUTINE ICAO
C      SUBROUTINE ICAO DETERMINES THE ATMOSPHERIC TEMPERATURE, PRESSURE, AND
C      DENSITY AS A FUNCTION OF ALTITUDE ABOVE THE EARTH IN ACCORDANCE WITH
C      THE 1962 U.S. STANDARD ATMOSPHERE (ICAO TO 20 KM.). A SHORT FAP
C      PROGRAM FOLLOWS ICAO WHICH PROVIDES A MEANS OF LOADING DATA INTO MACHINE.
C      IT MUST BE LOADED DIRECTLY AFTER ICAO. IF THE LENGTH OF ICAO IS CHANGED,
C      THE DATA MUST BE RELOCATED.
C
C      R IS DISTANCE TO CENTER OF EARTH IN METERS.
C      ALT IS VEHICLE ALTITUDE ABOVE EARTH IN METERS.
C      TABLE H IS METERS OF ALTITUDE FROM THE EARTH'S SURFACE AND IS
C      THE ARGUMENT OF ATMOSPHERE PROPERTY TABLE.
C      ALM IS THE MEAN SLOPE OF THE TABLE H VS. TM CURVE AT TABLE H.
C      TMR IS TM AT TABLE H.
C      REF P IS THE PRESSURE IN MILLIBARS AT TABLE H.
C      TM IS THE TEMPERATURE TIMES STD. MOLECULAR WEIGHT / ACTUAL
C      MOLECULAR WEIGHT. DEGREES KELVIN.
C      PRESS IS PRESSURE IN MILLIBARS.
C      DENSITY IS DENSITY IN KILOGRAMS PER CUBIC METER.
C      HEIGHT IS EITHER GEOPOTENTIAL ALTITUDE OR GEOMETRIC ALTITUDE IN METERS.
C
      COMMON C
C      DIMENSION A(600), B(700), C(4000),
1      TABLEH(23), TMR(23), REFP(23), ALM(23), RB(3)
C
      EQUIVALENCE
1(A,C ( 11)),(ALT,A ( 4)),(B,C (1111)),
2(DENSITY,B ( 29)),(OBLATN,A ( 40)),
3(PRESS,B ( 33)),(R,B ( 102)),(RB,B ( 193)),
4(RE,A ( 25)),(TABLT,B ( 20)),(TM,B ( 34)),
5(RESQRD,B ( 7))
      EQUIVALENCE (TABLEH(24),TMR),(TABLEH(47),ALM),(TABLEH(70),REFP)
C
      IF (OBLATN) 102,101,102
101 ALT = R - RE
      GO TO 103
102 ALT = R-6356783.28/SQRTF(.9933065783+.006693421685(RB(3)/R)**2)
103 IF (ALT-90000.) 105,104,104
104 HEIGHT = ALT
      GO TO 106
105 HEIGHT = ALT/(1.0+ALT/6356766.)
106 K=K
C
C      FIND THE HEIGHT IN A TABLE OF BASE DATA. DATA ARE
C      ARRANGED IN DECENDING ALT WITH 21 REGIONS. ABOVE THAT, PRESSURE AND
C      DENSITY ARE SET = 0. TEMPERATURE IS SET TO 3000.
      1 IF (K-22) 2,6,6
      2 IF (HEIGHT-TABLEH(K+1)) 5,3,3
      3 K = K+1
      GO TO 1
      4 K = K-1
      5 IF (K) 7,7,6
      6 HINC = HEIGHT - TABLEH(K)
      IF (H INC) 4,8,8
      7 K = 1
      8 IF (ALM(K)) 9,100,9
C

```



```

C      CONTROL COMES HERE FOR NONISOTHERMAL LAYERS
9      TM = TMR(K) + ALM(K)*H INC
      IF (ALT-90000.) 107,107,108
107     PRESS = REFP(K)*(TMR(K)/TM)**(.0341631947/ALM(K))
      GO TO 10
108     IF (K-KC) 109,110,109
109     KC = K
      C1 = RE+TABLEH(K)
      C2 = TMR(K)/ALM(K)
      C3 = 1./(C1-C2)
      C4 = -.0341631947*RESORD*C3/ALM(K)
110     PRESS = REFP(K)*EXPF(C4*(C3*LOGF(C1*(HINC/C2+1.)/(RE+HEIGHT))-
1      HINC/C1/(RE+HEIGHT)))
10     DENSITY = PRESS/TM/2.87053072
      GO TO 13

C
C      CONTROL COMES HERE FOR ISOTHERMAL LAYERS
100     IF (K-22) 11,12,12
11     TM = TMR(K)
      PRESS = REFP(K)*EXPF(-.0341631947*HINC/TMR(K))
      GO TO 10

C
C      CONTROL COMES HERE FOR EXTREME ALTITUDES
12     PRESS = 0.0
      DENSITY = 0.0
      TM = 3000.
13     RETURN
      END

      REM      THIS IS THE FAP PROGRAM WHICH LOADS ICAD DATA INTO MACHINE.
      REM      THE 256 IN ORG 256 WAS FOUND BY SUBTRACTING 22 FROM THE DEC LOCATION
      REM      OF REF P (FROM FAP LISTING OF ICAD, THIS WAS FOUND TO BE 278).
      REM      THUS, 278-22=256. DISCARD THE FIRST TWO BINARY CARDS AFTER ASSEMBLY
      REM      AND PLACE REMAINING CARDS IMMEDIATELY BEHIND ICAD BINARY DECK.
      REM
      REM      A1 IS REF P(23)
      REM      A2 IS ALM(23)
      REM      A3 IS TMR(23)
      REM      A4 IS TABLE H(23)
      REM
      ORG 256
A1     DEC 0.,1.1918E-9,3.4502E-9,1.0957E-8,4.0304E-8,1.8838E-7
      DEC 6.9604E-7,1.6852E-6,2.7926E-6,3.6943E-6,5.0617E-6,2.5217E-5
      DEC 7.3544E-5,3.0075E-4,1.6438E-3,.010377,.182099,.590005
      DEC 1.10905,8.68014,54.7487,226.320,1013.25
A2     DEC 0.,0.,.0011,.0017,.0026,.0033,.004,.005,.007,.01,.015,.02,.01
      DEC .005,.003,0.,-.004,-.002,0.,.0028,.001,0.,-.0065
A3     DEC 0.,2700.65,2590.65,2420.65,2160.65,1830.65,1550.65,1350.65
      DEC 1210.65,1110.65,960.65,860.65,760.65,660.65,560.65,460.65
      DEC 252.65,270.65,270.65,228.65,216.65,216.65,288.15
A4     DEC 1E30,7E5,6E5,5E5,4E5,3E5,2.3E5,1.9E5,1.7E5,1.6E5,1.5E5,1.2E5
      DEC 1.1E5,1E5,.9E5,79000.,61000.,52000.,47000.,32000.,20000.
      DEC 11000.,0.
      END

```

1

SUBROUTINE NBODY

```

C      NBODY COMPUTES THE TRAJECTORY IN EITHER ORBIT ELEMENTS OR RECTANGULAR
C      COORDINATES USING THE RUNGE-KUTTA TECHNIQUE. A LOWER ORDER INTEGRATION
C      TECHNIQUE IS ALSO PERFORMED TO FACILITATE AUTOMATIC STEP SIZE CONTROL.
C      THE X,XPRIM,XDOT,XINC,ETC. ARRAYS ARE AS FOLLOWS.

```

	X	ORBIT ELEMENTS	RECTANGULAR COORDINATES
1		TIME	TIME
2		MASS	MASS
3		ECCENTRICITY	X-VELOCITY
4		ARGUMENT OF PERICENTER	Y-VELOCITY
5		ARGUMENT OF ASC. NODE	Z-VELOCITY
6		INCLINATION	X
7		MEAN ANOMALY	Y
8		SEMI-MAJOR AXIS	Z

```

C      IMODE      VARIABLES
C      1      ORBIT ELEMENTS
C      2      RECTANGULAR
C      3      RECTANGULAR TEMPORARY
C      4      EARTH SPHERICAL--CHANGE TO RECTANGULAR
C      -1     ORBIT ELEMENTS--CHANGE TO RECTANGULAR
C      -2     RECTANGULAR--CHANGE TO ORBIT ELEMENTS
C      -3     ORBIT ELEMENTS--CHANGE TO TEMPORARY RECTANGULAR
C      -4     EARTH SPHERICAL -- CHANGE TO ORBIT ELEMENTS

```

COMMON C

```

C      DIMENSION A(600), B(700), C(4000),
1      XPRIM (100,2), XPRIMB (100,2), XDOTPM (100,2),
2      X (100), XINC (100), OLDINC (100),
3      XDOT (100), RB (3), XK (100),
4      AMC (3), AK (3), AW (4),
5      XWHOLE (6), VX (3), BEX (14)

```

```

EQUIVALENCE
1(A ,C ( 11)),(A1 ,B ( 10)),(A2 ,B ( 11)),
2(ACDEF1,B ( 12)),(ACDEF2,B ( 13)),(ACDEF3,B ( 14)),
3(AK ,A ( 51)),(AMC ,B ( 87)),(AMSGRD,B ( 91)),
4(AM ,B ( 90)),(ASYMPT,A ( 7)),(AW ,A ( 55)),
5(B ,C (111)),(CONSTU,A ( 32)),(DELT ,B ( 1)),
6(DONE ,B ( 39)),(E2 ,B ( 18)),(EMONE ,B ( 28)),
7(ERLIMT,A ( 14)),(ETOL ,A ( 30)),(EXMODE,B ( 27)),
8(GK2M ,B ( 36)),(H2 ,B ( 15)),(IMODE ,A ( 1)),
9(INDERR,B ( 51)),(KSUB ,B ( 19)),(MBODYS,B ( 42))
EQUIVALENCE
1(NEQ ,A ( 2)),(NSTART,B ( 24)),(OLDOEL,B ( 9)),
2(QMAX ,B ( 44)),(RATIO ,B ( 58)),(RB ,B ( 193)),
3(REVS ,A ( 48)),(R ,B ( 102)),(STEPMX,A ( 16)),
4(STEPMGO,A ( 41)),(STEPNO,A ( 42)),(TRANSFER,B ( 8)),
5(TRU ,B ( 40)),(TTEST ,A ( 54)),(VSQRD ,B ( 96)),
6(VX ,B ( 92)),(XDOT ,B ( 501)),(XINC ,B ( 601)),
7(XPRIM ,C ( 71)),(XPRIMB,C ( 911)),(XWHOLE,B ( 110)),
8(X ,B ( 40)),(ERLOG ,B ( 17)),(EREF ,A ( 13)),
9(Q ,B ( 59)),(OUTPUT,B ( 399))
C
C PART 1. SET UP THE STARTING SEQUENCE FOR ERROR CONTROL AND DELAY CHECKING
C THE ERROR UNTIL TWO STEPS ARE COMPLETED. THE ASSIGNED GO TOS NSTART AND
C TBEGIN CONTROL STARTING.
C NEQ = NEQ
1 DO 2 J=1,NEQ
XPRIM(J,2) = XPRIM(J,1)
XPRIMB(J,2) = XPRIMB(J,1)
2 X(J) = XPRIM(J,1)
NSTART = 0
TRANSFER = 0.
H2 = DELT
DELT = DELT/2.
220 CALL EQUATE
IF (OUTPUT) 222,221,222
221 CALL OUTPUT
222 DO 3 J=1,3
XWHOLE(J)=VX(J)
3 XWHOLE(J+3) = RB(J)
C CHANGE INTEGRATION VARIABLES IF IMODE IS -.
IF (IMODE) 4,5,5
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (TRANSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART 7A.
IF (RB(2)) 7,6,8
6 IF (VX(2)) 7,8,8
7 ASSIGN 37 TO NREV1
ASSIGN 35 TO NREV2
GO TO 9
8 ASSIGN 33 TO NREV1
ASSIGN 37 TO NREV2
9 DO 10 J=1,NEQ
XDOTPM(J,1) = XDOT(J)
XINC(J) = 0.
10 CONTINUE
11 KSUB = 1
ASSIGN 16 TO N
C
C PART 2. RUNGE-KUTTA SUBINTERVAL SCHEME. EQUATE PRODUCES THE NECESSARY
C DERIVATIVES XDOT(J).
C
12 DO 13 J=1,NEQ
XK(J) = XDOT(J) * DELT
XINC(J) = XINC(J) + AW(KSUB)*XK(J)
13 X(J) = XPRIM(J,2) + AK(KSUB)*XK(J)
14 CALL EQUATE
15 GO TO N,(16,17,18,20)
C
C PART 3. SUBINTERVALS 2, 3, AND 4, TO STATEMENT 19 FINISH A
C RUNGE-KUTTA STEP AND INCREMENT XPRIM(J,2) IN DOUBLE PRECISION.
C
16 KSUB = 2
ASSIGN 17 TO N
GO TO 12
17 KSUB = 3
ASSIGN 18 TO N
GO TO 12
18 DO 19 J=1,NEQ
XINC(J) = XINC(J) + AW(4) *XDOT(J) * DELT
180 CALL EXADD(XPRIM(J,2), XPRIMB(J,2), XINC(J))
X(J) = XPRIM(J,2)
19 CONTINUE
C
C PART 4. BEGIN A NEW RUNGE-KUTTA STEP. THIS ALSO GIVES DERIVATIVES
C FOR THE LOWER ORDER INTEGRATION CHECK.
C
ASSIGN 20 TO N
GO TO 14
20 GO TO NSTART,(27,23,21)
C
C PART 5. STARTING PHASE PROGRAM.
C PART 5A. THIS SECTION COMPLETES THE FIRST STEP OF STARTING PHASE.
C
21 ASSIGN 23 TO NSTART
DO 22 J=1,NEQ
OLDINC(J)=XINC(J)
XINC(J)=0.
XDOTPM(J,2) = XDOT(J)
22 CONTINUE
GO TO 11
C

```

```

C      PART 5B. MAX ERROR TEST--STARTING ONLY--CHECK THE MAX ERROR AND
C      EITHER ENTER RUNNING MODE OR REPEAT START WITH SMALLER STEP.
23 DO 24 J=2,NEQ
24 XINC(J) =(XINC(J)+OLDINC(J))*3.-(XDOTPM(J,1)+XDOTPM(J,2)*4.
1+XDOT(J))*DELT
240 CALL ERRORZ
25 IF (E2-ERLIMT) 26,26,56
26 ASSIGN 27 TO NSTART
  ASSIGN 11 TO IBEGIN
  A1 = A2
  GO TO 31
C
C      PART 6. RUNNING PHASE PROGRAM.
C      PART 6A. CHECK THE INTEGRATION BY INTEGRATING OVER THE LAST
C      RUNGE KUTTA STEP BUT USE DOTS FOR LAST TWO INTERVALS, OLDDDEL
C      AND DELT RESPECTIVELY. STATEMENT 28 IS THE LOWER INTEGRATION
C      MINUS RUNGE-KUTTA INCREMENTS. ERRORZ COMPUTES THE MAXIMUM RELATIVE
C      ERROR AND STATEMENT 29 TESTS THIS ERROR AGAINST THE LIMIT VALUE.
27 RATIO = DELT/OLDDDEL
  HFACT=DELT/(1.+RATIO)
  ACOEF1=-RATIO*RATIO*HFACT
  ACOEF2=RATIO*(DELT+3.*OLDDDEL)
  ACOEF3=DELT+DELT*HFACT
  DO 28 J=2,NEQ
28 XINC(J) = ACOEF1*XDOTPM(J,1)+ACOEF2*XDOTPM(J,2)-6.*XINC(J)
1+ACOEF3*XDOT(J)
280 CALL ERRORZ
29 IF (E2-ERLIMT) 30,30,57
C
C      PART 7A. LAST POINT OKAY. COUNT THE REVOLUTIONS PAST THE X-AXIS.
C      A STEP GREATER THAN 1/2 REV. MAY FAIL TO ADD IN.
30 H2 = DELT
31 QMAX = MAX1F(Q,QMAX)
  IF (RB(2)) 32,34,36
32 GO TO NREV1, (37,33)
33 ASSIGN 37 TO NREV1
  ASSIGN 35 TO NREV2
  GO TO 37
34 GO TO NREV2, (37,35)
35 ASSIGN 33 TO NREV1
  ASSIGN 37 TO NREV2
36 REVS = REVS + 1.
37 IF (XABSF(IMODE)-1) 42,38,42
C
C      PART 7B. IN ORBIT ELEMENTS. ADJUST ARGUMENT OF PERICENTER AND MEAN ANOMALY
C      TO + OR - PI TO MAINTAIN ACCURACY IN SIN-COS ROUTINES.
38 IF (EMONE) 39,42,42
39 DO 41 J=4,7,3
  ADJ2=INTF(XPRIM(J,2)/6.28318532+SIGNF(.5,XPRIM(J,2)))
  IF (ADJ2) 40,41,40
40 ADJ3 = -ADJ2*6.28125
400 CALL EXADD(XPRIM(J,2),XPRIMB(J,2),ADJ3)
  ADJ3=-ADJ2+.0019353072
401 CALL EXADD(XPRIM(J,2),XPRIMB(J,2),ADJ3)
41 CONTINUE
C
C      PART 7C. ADVANCE THE REMAINING PARAMETERS, FIND NEW STEP SIZE,
C      AND TEST FOR AN ORIGIN TRANSLATION.
42 DO 43 K=1,3
  XWHOLE(K)=VX(K)
43 XWHOLE(K+3) = RB(K)
  DO 44 J=1,NEQ
  XDOTPM(J,1) = XDOTPM(J,2)
  XDOTPM(J,2) = XDOT(J)
  XPRIM(J,1) = XPRIM(J,2)
  XPRIMB(J,1) = XPRIMB(J,2)
  XINC(J) = 0.
44 CONTINUE
  OLDDDEL = DELT
45 CALL STEP
  IF (IDONE) 67,450,67
450 IF (NSTART) 451,1,451
451 IF (MBDDYS) 46,47,46
46 CALL TESTTR
  IF (TRSFER) 1,47,1
47 IF (XABSF(IMODE)-3) 11,48,11
C
C      PART 7D. IF IN TEMPORARY RECTANGULAR COORDINATES, TEST FOR RETURN
C      TO ORBIT ELEMENTS. FIRST, E IS FOUND. IF TIME HAS NOT ADVANCED
C      SUFFICIENTLY, INTEGRATION CONTINUES IN RECTANGULAR VARIABLES (STATE. 48).
C      STATEMENT 49 DETERMINES IF KEPLERS EQUATION CAUSED IMODE = 3. IF NOT,
C      AN E CLOSE TO 1 CHECK IS MADE IN STATEMENT 50. IF IT DID, RECTANGULAR
C      VARIABLES WILL BE USED IF THE LIMIT IS TOO SMALL (STATEMENT 52), OR
C      IF E IS 5 OR GREATER (STATEMENT 53) OR IF THE PATH LIES CLOSE TO AN
C      ASYMPTOTE (STATEMENT 55).
48 CALL CONVT1 (VX,AMC)
  EXMODE=SQRTF(1.+AMSQRD/GK2M*(VSQRD/GK2M-2./R))
  EMONE=EXMODE-1.
  IF ((XPRIM(1)-TTEST)*DELT) 11,11,49
49 IF (ASYMPT) 51,50,51
50 IF (ETOL-ABSF(EMONE)) 55,11,11
51 IF (EMONE) 55,55,52
52 IF (CONSTU-1.E-7) 11,53,53
53 IF (EXMODE-5.) 54,11,11
54 CALL CONVT2
  IF (ABSF(STRU)-2.2/SQRTF(EXMODE)) 55,55,11
55 ASYMPT = 0.0
  IMODE=-2
555 CALL TESTTR
  GO TO 1
C

```

```

C      PART 8. COMES HERE WHEN ERROR TEST FAILED--BOTH STARTING AND RUN.
C      RETRIEVE OLD POINT AND RECOMPUTE WITH SMALLER INTERVAL.
C      IF TWO CONSECUTIVE TRY'S FAIL (STATEMENT 59) THE STARTING SEQUENCE OCCURS.
56  ASSIGN 1 TO IBEGIN
57  DO 58 J=1,NEQ
      XPRIM(J,2) = XPRIM(J,1)
      XPRIMB(J,2) = XPRIMB(J,1)
      XDOT(J)=XDOTPM(J,2)
      XINC(J)= 0.
58  CONTINUE
      STEPNO=STEPNO+1.
      HZ = DELT
      DELT=SIGNF(EXPF((ERLOG-A2)/5.),DELT)
      A2 =A1
59  IF (FAIL-STEPGO) 60,61,60
60  FAIL = STEPGO
      GO TO IBEGIN, (11,1)
61  ASSIGN 1 TO IBEGIN
      IF (STEPNO + STEPGO - STEPMX) 62,62,45
62  GO TO IBEGIN, (11,1)

C
C      PART 10. PRINT OUT THE ERROR INFO. IF EREF HAS A - SIGN. THEN RETURN.
67  IF (EREF) 68,72,72
68  WRITE OUTPUT TAPE 6,70
      REWIND 4
      DO 69 I=1,INDERR
        READ TAPE 4, BEX
69  WRITE OUTPUT TAPE 6,71,BEX
      REWIND 4
      INDERR = 0
70  FORMAT(7H1 STEP,6X,4HTIME,6X,4HDELT,7X,2HA2,8X,2HE2,7X,4HMASS,6X,
14HE,VX,4X,8HOMEGA,VY,2X,8HNUDES,VZ,3X,6HINCL,X,5X,4HMA,Y,6X,3HP,Z,
24X,1HK//)
71  FORMAT(F5.,1H+F3.,1P11G10.2,I2)
72  RETURN
END

C      SUBROUTINE ORDER
C      THIS ROUTINE TAKES THE BODY LIST READ FROM CARDS AND SORTS THEM IN
C      ORDER SO THAT THE DISTANCE FROM THE REFERENCE TO EACH BODY IS
C      DEPENDENT UPON ALREADY COMPUTED DISTANCES ONLY.
C
C      ELLIPSE DATA ARE READ INTO A BLOCK OF 120 STORES RESERVED FOR
C      TEN ELLIPSES. ONE ELLIPSE IS READ INTO A 12 STORE BLOCK.
C      THE SINES AND COSINES OF THE 3 ANGLES ARE COMPUTED AND STORED
C      IN THE TDATA ARRAY ALONG WITH THE REST OF THE ELLIPSE DATA.
C      A BLOCK IS ARRANGED AS FOLLOWS:
C
C      (1) = NAME OF BODY IN BCD, ONLY 6 CHARACTERS.
C      (2) = NAME OF REFERENCE BODY IN BCD, SAME RESTRICTION.
C      (3) = MASS OF THE BODY IN SUN MASS UNITS.
C      (4) = RADUIS INSIDE OF WHICH COORDINATES WILL BE TRANSLATED TO THIS BODY.
C      (5) = SEMILATUS RECTUM IN ASTRONOMICAL UNITS.
C      (6) = ECCENTRICITY OF THE ORBIT.
C      (7) = LONGITUDE OF ASCENDING NODE.
C      (8) = ARGUMENT OF PERIHELION.
C      (9) = INCLINATION OF THE ORBIT.
C      (10) = PERIGEE PASSAGE JULIAN DAY.
C      (11) = PERIGEE PASSAGE FRACTION OF DAY.
C      (12) = PERIOD OF THE ELLIPSE IN MEAN SOLAR DAYS.
C
C      AMASS = MASS OF EACH BODY, SUN MASSES. ORDER OF PNAME.
C      BMASS = SELECTED FROM AMASS. CORRESPONDS TO BNAME LIST.
C      BNAME = THE ORDERED LIST OF BCD BODY NAMES. CAN BE USED IN OUTPUT.COMMON.
C      BODYCD = THE ORIGINAL BCD NAMES READ FROM CARDS.
C      BODYL = THE LIST OF BCD BODY NAMES WITH THE REFERENCE BODY AT TOP.
C      INITIALLY EQUAL TO BODY CARD LIST (BODYCD).
C      IBODY = ARRAY OF SUBSCRIPTS. WHEN A DISTANCE IS FOUND FROM EPHEMERIS, IT
C      MAY BE ADDED (OR SUBTRACTED) FROM THE BODY POSITION GIVEN BY
C      XP(IBODY) TO OBTAIN THE POSITION OF THE PRESENT BODY. COMMON.
C      KZERO = COUNT OF ZERO REFERENCES. THERE MUST BE ONE AND ONLY ONE ZERO.
C      FROM LOCATION IN BNAME LIST. NOT IN COMMON.
C      MANE = ARRAY OF SUBSCRIPTS. INVERSE OF NAME. GIVES NEW LOCATION OF
C      BNAME LIST IN TERMS OF BODYL. NOT IN COMMON.
C      NBODYS = COUNTED INTERNALLY. TOTAL NUMBER OF BODYS.
C      MBODYS = COMPUTED INTERNALLY. TOTAL NUMBER OF EPHEMERIDES (NBODYS-1).
C      NAME = ARRAY OF SUBSCRIPTS. GIVES OLD LOCATION OF NAMES IN BODYL
C      NEFMRS = ARRAY OF SUBSCRIPTS. GIVES LOCATION OF BODY IN PNAME LIST
C      IN TERMS OF THE EFMRS LIST. STORED IN COMMON.
C      NREFER = ARRAY OF SUBSCRIPTS. LOCATES THE REFERENCE BODY IN BODYL.
C      ORDER OF THE ARRAY CORRESPONDS TO BODYL. NOT IN COMMON.
C      NNREFR = ARRAY OF SUBSCRIPTS. LIKE NREFER BUT REFERS AND CORRESPONDS TO
C      BNAME LIST. NOT IN COMMON.
C      PNAME = A PERMANENT LIST OF BCD BODY NAMES. 1 WORD EACH (6 CHARACTERS
C      MAX). USED TO IDENTIFY MASS, REFERENCE NAMES, ETC. THE LIST IS
C      A MAXIMUM OF 30 NAMES. PRECISION TAPE NAMES ARE FROM 1 TO 20,
C      ELLIPTIC NAMES ARE FROM 21 TO 30.
C      REFER = A PERMANENT LIST OF BCD BODYS THAT ARE THE REFERENCES OF
C      DISTANCES GIVEN IN EPHEMERIDES (TAPES OR ELLIPSE). CORRESPONDS
C      TO PNAME LIST.
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000),
1  AMASS (30), BMASS (8), BNAME (8),
2  BODYL (8), EFMRS (7), IBODY (8),
3  MANE (8), NAME (8), NEFMRS (8),
3  NEFMRT (8), NNREFR (8), BODYCD (8),
4  NREFER (8), PNAME (30), RBCRIT (7),
5  RCRIT (30), REFER (30), TDATA (18,7),
6  TDEL (7), TIM (7), ELIPS (120),
7  NDUD (9), XPRIM (200)

```

```

EQUIVALENCE
1(A ,C ( 11)),(AMASS ,A ( 347)),(ATMN ,A ( 21)),
2(AU ,A ( 291)),(B ,C (1111)),(BMASS ,B ( 137)),
3(BNAME ,B ( 122)),(BODYCD,A ( 143)),(BODYL ,B ( 153)),
4(EFMR ,B ( 130)),(ELIPS ,A ( 167)),(FILE ,B ( 22)),
5(GK2M ,B ( 36)),(GKM ,B ( 37)),(IBODY ,B ( 177)),
6(NBODY ,B ( 42)),(NBODYS ,B ( 41)),(NEFMRS ,B ( 185)),
7(PNAME ,A ( 287)),(RATMOS ,B ( 23)),(RATH ,A ( 22)),
8(RBCRIT ,B ( 145)),(RCRIT ,A ( 377)),(REFER ,A ( 317)),
9(IRE ,A ( 25)),(RESQRD ,B ( 7)),(REVOLV ,B ( 21))
EQUIVALENCE
1(ROTATE ,A ( 39)),(SPD ,A ( 44)),(SQROK1 ,A ( 47)),
2(SQROK ,B ( 35)),(TDATA ,B ( 265)),(TDEL ,B ( 170)),
3(TFILE ,A ( 6)),(TIM ,B ( 163)),(TRANSFER ,B ( 8)),
4(XPRIM ,C ( 71)),(OUTPOT ,B ( 399))
EQUIVALENCE (MANE(1),NOUD(2))
C
C THIS SECTION SEES WHAT ELLIPSE DATA WAS READ FROM CARDS AND PUTS THE
C NAMES IN PLACE SO THAT DATA WILL BE USED IF NEEDED. ELLIPSE DATA HAS
C PRIORITY OVER TAPE DATA BECAUSE LAST DATA IN LIST IS THAT ACTUALLY USED.
C FUNCTION COMPARF(A,B) IS EQUIVALENT TO (A-B) BUT WILL NOT OVERFLOW.
C
C COMPARF(A,B) = (A+B)*(-(A-B))
C DO 2 K=1,120,12
C IF (ELIPS(K)) 1,2,1
C 1 KOUNT = (K-1)/12+21
C PNAME(KOUNT) = ELIPS(K)
C REFER(KOUNT) = ELIPS(K+1)
C AMASS(KOUNT) = ELIPS(K+2)
C RCRIT(KOUNT) = ELIPS(K+3)
C 2 CONTINUE
C
C PART 0. THROW AWAY BLANKS AND DUPLICATES IN BNAME LIST.
C ALSO COUNT THE BODIES.
C IF (TRANSFER) 4,3,4
C 3 BNAME(1) = BODYCD(1)
C 4 DO 5 K=1,8
C 5 BNAME(K+1) = BODYCD(K)
C L = 1
C BODYL(0) = 0.
C DO 8 I=1,9
C BODYL(I) = 0.
C DO 6 K=1,L
C IF (COMPARF (BNAME(I), BODYL(K-1))) 6,7,6
C 6 CONTINUE
C BODYL(L) = BNAME(I)
C L = L+1
C 7 BNAME(I) = 0.
C 8 CONTINUE
C NBODYS = L-1
C MBODYS = NBODYS-1
C
C PART 1. FIND THE REFERENCE BODY FOR EACH BODY IN THE LIST OF BODYS
C READ FROM CARDS. CLEAR NREFER AND BNAME.
C DO 13 KL=1,NBODYS
C NREFER(KL) = 0
C NEFMRT(KL) = 0
C BNAME (KL) = 0.
C DO 12 KP= 1,30
C IF (COMPARF(BODYL(KL),PNAME(KP))) 12,9,12
C 9 NEFMRT(KL) = KP
C DO 11 KR = 1,8
C IF (COMPARF(REFER(KP),BODYL(KR))) 11,10,11
C 10 NREFER(KL) = KR
C 11 CONTINUE
C 12 CONTINUE
C 13 CONTINUE
C PART 2. COUNTS 0 REFERENCES AND SAVES TEMPORARY SET OF INDEXES.
C 14 IF (NBODYS) 24,24,15
C 15 KZEROS = 0
C MISPEL = 0
C DO 20 K = 1,NBODYS
C NNREFER(K) = NREFER(K)
C 16 IF (NEFMRT(K)) 18,17,18
C 17 MISPEL = MISPEL + 1
C 18 IF(NREFER(K)) 20,19,20
C 19 KZEROS = KZEROS + 1
C 20 CONTINUE
C 21 IF (KZEROS- 1) 24,22,24
C 22 IF (MISPEL) 24,23,24
C 23 IF (NBODYS-8) 28,28,24
C
C PART 3. REPORTS ERRORS IN BODY LIST.
C 24 WRITE OUTPUT TAPE 6,25 ,NBODYS,MISPEL,KZEROS,(BODYL(K),K=1,NBODYS)
C WRITE OUTPUT TAPE 6,26 ,(NREFER(K),K=1,NBODYS)
C WRITE OUTPUT TAPE 6,27 ,(K,PNAME(K),REFER(K),K=1,30)
C 25 FORMAT (26H000FY BODY LIST (NBODYS =12,13H, MISSPELL =12,
C 11H, KZEROS =12,1H)/11H0BODYLST =8(3X,A6))
C 26 FORMAT (11H NREFER =16,719)
C 27 FORMAT (/5(13H K3X,4HBODY4X,5HREFER5X,)/5(13,2X,A6,2X,A6,5X))
C CALL EXIT
C
C PART 4. TRACES OUT ..REFERENCE TO BODY.. RELATIONSHIPS
C 28 KK = 2
C KN = 1
C NAME(1) = 1
C 29 IF (NREFER(KN)) 24,31,30
C 30 NAME(KK) = NNREFER(KN)
C NNREFER(KN) = 0
C KN = NAME(KK)
C KK = KK + 1
C GO TO 29
C

```

```

C   PART 5.  TRACES OUT ..BODY TO REFERENCE..  RELATIONSHIP
31 DO 34 KN = 1,NBODYYS
   DO 34 K = 1,NBODYYS
32 IF (NNREFR(K) - NAME(KN)) 34,33,34
33 NAME(KK) = K
   KK = KK + 1
34 CONTINUE

C
C   PART 6.  INVERTS NAME TO MANE,STORES BNAME, BMASS, RBCRIT,  AND A
C   TEMPORARY NEFMRS.
   DO 35 K = 1,NBODYYS
   N = NAME(K)
   MANE(N) = K
   NEF = NEFMRT(N)
   BNAME(K) = PNAME(NEF)
   BMASS(K) = AMASS(NEF)
   RBCRIT(K) = RCRIT(NEF)
   NEFMRS(K) = NEF
35 CONTINUE

C
C   PART 7.  FINDS NNREFR REFERENCE FOR BNAME LIST , ALSO TEMP. IBODY
   DO 36 K = 1, NBODYYS
   N = NAME(K)
   NRF = NREFER(N)
   NNREFR(K) = MANE(NRF)
36 IBODY(K) = MANE(NRF)

C
C   PART 8 . FINDS IBODY FOR BACKWARD REFERENCE.
   DO 39 K=1,8
37 IF(NNREFR(K)) 24,40,38
38 N = NNREFR(K)
   IBODY(N) = -K
39 CONTINUE
   IBODY LIST IS COMPLETE.

C
C   PART 9 . WRITES OUT EPHEMERIS LIST TO BE USED IN STORING DATA AND
C   MAKES FINAL NEFMRS LIST.
   40 KK = 1
   DO 43 K=1,NBODYYS
   41 IF(NNREFR(K)) 42,43,42
   42 EFMRS(KK) = BNAME(K)
   NEFMRS(KK) = NEFMRS(K)
   KK = KK + 1
   43 CONTINUE
   NEFMRS(NBODYYS) = 0

C
C   PART 10. SAVES ELLIPSE DATA
   FILE = 0.
   IF (MBODYYS) 430,480,430
430 DO 48 K=1,MBODYYS
   44 IF(NEFMRS(K)-20) 47,47,45
   45 DO 46 J=5,12
   L = (NEFMRS(K) - 21) * 12 + J
   46 TDATA(J-4,K) = ELIPS(L)
   DO 50 J=7,9
   L = (NEFMRS(K)-21)*12+J
   TDATA(J+2,K) = SIN(ELIPS(L))
   50 TDATA(J+5,K) = COS(ELIPS(L))
   GO TO 48

C
C   PART 10A.  LOADS A FALSE (VERY EARLY) TAPE TIME TO FORCE TAPE
C   READING BY THE EPHMRS ROUTINE.  FILE = 0 UNLESS TAPE IS USED.
   47 TDEL(K) = 0
   TIM(K) = 2400000.5
   FILE = 10.
   48 CONTINUE

C
C   PART 11. COMPUTE GRAVITATIONAL CONSTANTS. 1.9866 E+30 = KILOGRAMS/SUN MASS
C   IF ORIGIN BODY HAS AN ATMOSPHERE, SET ROTATION RATE AND ATMOSPHERE RADIUS.
C   POSITION THE EPHEMERIDES TAPE AT THE BEGINNING OF THE CORRECT EPHEMERIS
C   BY MATCHING THE EPHEMERIS NUMBER READ FROM TAPE (FILE) WITH THE DESIRED
C   EPHEMERIS NUMBER (IFILE).
   480 RESQD = RE**2
   SQDK = SQDK1*AU**3/SPD**2
   GK2M = SQDK*(BMASS(1) + XPRIM(2)/1.9866 E30)
   GKM = SQRTF(GK2M)
   REVOLV = 0.
   RATMOS = 0.
   IF (ATMN-BNAME(1)) 51,49,51
   49 REVOLV = ROTATE
   RATMOS = RATM
   51 IF (FILE) 56,56,52
   52 CALL BSFILE(3)
   53 READ TAPE 3, FILE
   IF (FILE-TFILE) 54,56,55
   54 CALL SKFILE(3)
   GO TO 53
   55 BACKSPACE 3
   BACKSPACE 3
   GO TO 52

C
C   PART 12.  WRITES THE BNAME LIST ON TAPE 6 .
   56 IF (OUTPOT) 58,59,58
   59 WRITE OUTPUT TAPE 6,57,BNAME(1),(BNAME(K),K=2,NBODYYS)
   57 FORMAT (19HOREFERENCE BODY IS A6,5X,23H PERTURBING BODIES ARE
   1 7(2X,A6))
   58 RETURN
   END

```

```

C                                     SUBROUTINE OBLATE
C THIS SUBROUTINE COMPUTES THE OBLATENESS ACCELERATIONS (OBLAT) DUE TO AN
C AXIALLY SYMMETRIC EARTH. THE 2ND, 3RD, AND 4TH SPHERICAL HARMONIC COEFF.
C ARE OBLATJ, OBLATH, AND OBLATD RESPECTIVELY.
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000),
C      1 RB(3), OBLAT(3)
C
C      EQUIVALENCE
C      1(A ,C ( 11)),(B ,C (1111)),(GK2M ,B ( 36)),
C      2(OBLATJ,A ( 26)),(OBLATD,A ( 27)),(OBLATH,A ( 28)),
C      3(OBLAT ,B ( 75)),(R ,B ( 102)),(RB ,B ( 193)),
C      4(RE ,A ( 25)),(RSQRD ,B ( 45)),(RESQRD,B ( 71))
C
C      AA = RB(3)/R
C      AB = AA*AA
C      IF (ABSF(AA)-1.E-6) 1,1,2
C      1 AA=0.
C      2 AB=0.
C      2 AC = RESQRD/RSQRD
C      AD = GK2M/RSQRD/R*AC
C      AE = OBLATJ*AD
C      AF = OBLATH*AD*RE/R
C      AG = OBLATD*AD*AC
C      AH = AE*(5.-AB-1.)+AF*(7.-AB-3.)*AA+AG*(6.-AB-9.-AB**2-0.4285714286)
C      OBLAT(1) = AH*RB(1)
C      OBLAT(2) = AH*RB(2)
C      OBLAT(3) = (AH-2.-AE+AG*(4.-AB-1.714285714))*RB(3)-AF*(3.-AB-0.6)*R
C      3 RETURN
C      END

```

```

C                                     SUBROUTINE OUTPUT
C ENTS AND RECTANGULAR COORDINATES ARE OUTPUTTED. IF THE OBJECT IS NOT WITH
C THIS IS THE ROUTINE WHICH FORMS THE BASIC DATA OUTPUT. BOTH ORBIT ELEM-
C IN AN ATMOSPHERE (PRESS=0.), ONE LINE OF DATA IS DELETED. LIKEWISE,
C ONLY THOSE PERTURBING BODIES PRESENT HAVE THEIR DISTANCES OUTPUTTED.
C
C      COMMON C
C
C      DIMENSION A(600), B(700), C(4000),
C      1 R (8), ORBELS (6), VATM (3),
C      2 BNAME (8), RB(3,8), DIRCOS(3,8),
C      3 XPRIM (200), RAMC (5)
C
C      EQUIVALENCE
C      1(A ,C ( 11)),(ALPHA ,A ( 49)),(ALT ,A ( 4)),
C      2(AMC ,B ( 87)),(AM ,B ( 90)),(AREA ,B ( 6)),
C      3(BNAME ,B ( 122)),(B ,C (1111)),(CD ,A ( 165)),
C      4(ICL ,A ( 164)),(COSALF,B ( 48)),(COSTRU,B ( 53)),
C      5(DTOFFJ,A ( 23)),(H2 ,B ( 15)),(IMODE ,A ( 1)),
C      6(MBODYS,B ( 42)),(NBODYS,B ( 41)),(ORBELS,B ( 116)),
C      7(PRESS ,B ( 33)),(P ,B ( 84)),(PSI ,B ( 30)),
C      8(PSIR ,B ( 398)),(PUSH ,A ( 166)),(Q ,B ( 59)),
C      9(RAMC ,B ( 393)),(RB ,B ( 193)),(REVS ,A ( 48))
C
C      EQUIVALENCE
C      1(R ,B ( 102)),(SINALF,B ( 46)),(SINTRU,B ( 52)),
C      2(STEPGO,A ( 41)),(STEPNO,A ( 42)),(TABLT ,B ( 20)),
C      3(TRU ,B ( 40)),(VATM ,B ( 97)),(VQ ,B ( 100)),
C      4(V ,B ( 95)),(VX ,B ( 92)),(VY ,B ( 93)),
C      5(VZ ,B ( 94)),(XPRIM ,C ( 71)),(OUTPUT,B ( 399))
C
C      DAYJ=(DTCFFJ-2.4E6)+TABLT
C      ALPHA1 = ALPHA*57.29577951
C      REV = REVS+ARCTAN(-RB(2),-RB(1))/6.28318532 + .5
C      16 CALL CONVT1(VX,AMC)
C      IMODE=IMODE
C      GO TO (2,1,1),IMODE
C      1 CODE=6HRECTAN
C      18 CALL CONVT 2
C      GO TO 4
C      2 DO 3 K=1,6
C      3 ORBELS(K) = XPRIM(K+2)
C      CODE=5HORBIT
C      TRU=ARCTAN(SINTRU,COSTRU)
C      4 PSI = ATANF((RB(1)*VX+RB(2)*VY+RB(3)*VZ)/AM)57.2957795
C      IF (OUTPUT) 19,6,19
C      6 WRITE OUTPUT TAPE 6, 11,STEPGO,STEPNO,ORBELS(1),ORBELS(2),V,R(1),B
C      1NAME(1),CODE,IMODE,XPRIM(1),ORBELS(6),TRU,VX,RB(1),XPRIM(2),DAYJ,D
C      2RBELS(5),ORBELS(3),VY,RB(2),REV,ALPHA1,PSI,ORBELS(4),VZ,RB(3),H2
C
C      IF WITHIN AN ATMOSPHERE COMPUTE DRAG, LIFT, G, ETC., AND PRINT EXTRA LINE.
C      19 IF (PRESS) 5,7,5
C      5 XIFT = Q*AREA*CL
C      DRAG = Q*AREA*CD
C      G = (PUSH-DRAG*COSALF+XIFT*SINALF)/XPRIM(2)/9.80665
C      17 CALL CONVT1(VATM,RAMC)
C      PSIR = ATANF((RB(1)*VATM(1)+RB(2)*VATM(2)+RB(3)*VATM(3))/RAMC(4))*
C      1 57.2957795
C      IF (OUTPUT) 7,14,7
C      14 WRITE OUTPUT TAPE 6,12,ALT,PSIR,DRAG,VQ,G,PUSH
C

```

```

C      IF PERTURBATING BODIES ARE PRESENT, FIND THEIR DISTANCES AND PRINT THEM.
7      IF(NBODY5) 8,10,8
8      DO 9 J=2,NBODY5
9      DO 9 K=1,3
10     DIRCOS(K,J) = -RB(K,J)/R(J)
11     IF (OUTPOT) 10,15,10
15     WRITE OUTPUT TAPE 6,13,
16     I(BNAME(J),R(J),DIRCOS(1,J),DIRCOS(2,J),DIRCOS(3,J),J=2,NBODY5)
10     RETURN
11     FORMAT(6HSTEP=F6.,2H +F4.,3X,13HECCENTRICITY=1PG15.8,7H OMEGA=G15
12     1.8,4H V=G15.8,3H R=G15.8,7H REFER=A6,1X,A6,12/6H TIME=1PG14.7,14
13     2H SEMILATUS R.=G15.8,7H TRU A=G15.8,4H VX=G15.8,3H X=G15.8,7H RMAS
14     35=G15.8/9H JDAY= 240PF10.4,15H MEAN ANOMALY=1PG15.8,7H NODE=G15.
15     48,4H VY=G15.8,3H Y=G15.8,7H REVS.=G15.8/6H ALFA=G14.7,14H PATH A
16     5NGLE=G15.8,7H INCL=G15.8,4H VZ=G15.8,3H Z=G15.8,7H DELT=G15.8)
12     FORMAT(6H ALT.=1PG14.7,14H R PATH ANGLE=G15.8,7H DRAG=G15.8,4H VR
13     1=G15.8,3H G=G15.8,7H PUSH=G15.8)
13     FORMAT(2(1X,A6,3H R=1PG14.7,OP3F10.6,11X))
END

FUNCTION QUAD (X,IC)
C      THIS ROUTINE COMPUTES ANY VARIABLE, QUAD, AS A QUADRATIC FUNCTION OF X.
C      QUAD = A + BX + CX. THERE MAY BE SEVERAL SETS OF COEFFICIENTS, EACH SET
C      BELONGING TO A PARTICULAR REGION OF X. THE COEFN ARRAY IS ARRANGED AS --
C      X1,A1,B1,C1,X2,A2,B2,C2,X3,A3,B3,C3,X4, .....
C      WHERE A1,B1,C1 ARE THE COEFFICIENTS TO BE USED FOR X BETWEEN X1 AND X2, ETC.
C      AND X1 IS LESS THAN X2, X2 IS LESS THAN X3, X3 IS LESS THAN X4, ETC.
C      IC IDENTIFIES WHICH DEPENDENT VARIABLE, QUAD, IS BEING SOUGHT.
C      ICC(IC) DEFINE THE STARTING LOCATIONS IN THE COEFN ARRAY FOR VARIABLES X.

COMMON C
C      DIMENSION A(600), B(700), C(4000),
1      COEFN(190), ICC(5)
C
C      EQUIVALENCE
1(A ,C ( 11)),(B ,C (111)),(COEFN ,A ( 407)),
2(ICC ,A ( 153))
C
C      I=ICC(IC)
1 IF (X-COEFN(I)) 2,3,3
2 I = I-4
GO TO 1
3 IF(X-COEFN(I+4)) 5,5,4
4 I = I+4
GO TO 3
5 QUAD = COEFN(I+1)+X*(COEFN(I+2)+X*COEFN(I+3))
ICC(IC)=I
RETURN
END

SUBROUTINE STAGE
C      THIS ROUTINE IS CALLED TO PREPARE DATA FOR USE IN NBODY. STAGE DATA IS
C      TAKEN FROM PERMINENT STORES AND LOADED INTO WORKING STORES. STAGE DATA
C      MAY BE SET ASIDE FOR LATER USE (IF ON NSAVE-NSTAGE). WHEN IMODE IS 4,
C      CONVERSION FROM EARTH-SPHERICAL TO RECTANGULAR OR ORBIT ELEMENTS TAKES
C      PLACE IN TUDES.
C
COMMON C
C      DIMENSION A(600), B(700), C(4000),
1XPRIM(200),XPRIMB(200),TB(10),FLOW1(10),AEXIT1(10),SIMP1(10),
2AREAL(10),DELT1(10),IDENT(10),TABLE1(200),RMAS1(10),D(600)
C
C      EQUIVALENCE
1(A ,C ( 11)),(AEXIT ,B ( 3)),(AEXIT1,A ( 103)),
2(AREAL ,A ( 113)),(AREA ,B ( 6)),(B ,C (111)),
3(DELT1 ,A ( 133)),(DELT ,B ( 1)),(D ,C (211)),
4(DELT ,A ( 43)),(DELMAX,A ( 19)),(DONE ,B ( 39)),
5(EREF ,A ( 13)),(ERLOG ,B ( 17)),(EXITA ,B ( 392)),
6(FLOW ,B ( 5)),(FLOW1 ,A ( 83)),(IDENT ,A ( 123)),
7(IMODE ,A ( 1)),(LSTAGE,A ( 38)),(MODDOUT,A ( 20)),
8(NCASE ,C ( 1)),(NCASES,A ( 600)),(NSAVE ,C ( 4)),
9(INSTAGE,A ( 3)),(PUSHO ,B ( 39)),(RMAS1,A ( 73))
C      EQUIVALENCE
1(SIMP1 ,A ( 93)),(SIMP ,B ( 2)),(TB ,A ( 63)),
2(TABLE ,C (191)),(TKICK ,A ( 15)),(TMAX ,B ( 4)),
3(TTOL ,A ( 45)),(XPRIMB,C ( 91)),(XPRIM ,C ( 71)),
4(RETURN,B ( 400)),(OUTPOT,B ( 399))
C

```



```

C      PART 0. SAVE INITIAL DATA IF DESIRED.  LOAD STAGE DATA INTO WORKING
C      STORAGE.  ALLOW ADDITIONAL STAGE INPUT.
      IF (DEL) 100,99,100
99      DEL = DELMAX-TKICK
100     IF (NSAVE-NSTAGE) 103,101,103
101     NCASES = NCASE
      DO 102 J=1,1100
102     D(J) = A(J)
      IF (OUTPUT) 103,97,103
97      WRITE OUTPUT TAPE 6,98,NSTAGE,NCASE
98      FORMAT(29H SAVED INITIAL DATA FOR STAGE12,8H OF CASE14,1H.)
103     NSTAGE = NSTAGE
      TMAX = XPRIM(1)+TB(NSTAGE)
      XPRIM(2) = 0.
      IF (RMASS1(NSTAGE)) 117,117,118
117     XPRIM(2) = XPRIM(2)+RMASS1(NSTAGE)
      GO TO 119
118     XPRIM(2) = RMASS1(NSTAGE)
119     FLOW = FLOW1(NSTAGE)
      SIMP = SIMP1(NSTAGE)
      AEXIT = AEXIT1(NSTAGE)
      AREA = AREA1(NSTAGE)
      DELT = DELT1(NSTAGE)
      ID = IDENT(NSTAGE)
3      CALL INPUT (ID,C,TABLE)
      ERLOG = LOGF(ABSF(EREF))
      TTOL = 5E-8*ABSF(TMAX)+1E-8
      PUSHO = SIMP*FLOW*9.80665
      EXITA = AEXIT+100.
      MODOUT = MODOUT
      IF (DELT) 105,104,105
104     DELT = TB(NSTAGE)/100.
      DELT1(NSTAGE) = DELT
105     GO TO (109,106,106,109), MODOUT
106     IF (DEL-DELMAX) 108,108,107
107     DEL = MODF(DEL,DELMAX)
108     IF (DEL) 114,109,114
114     DELT = MIN1F(DELT,DEL)
109     IF (XABSF(IMODE)-4) 1,110,1
110     CALL TUDES
      IMODE = XSIGNF(2,IMODE)
1      CALL NBDY
2      CALL EXTRAS
C
C      PART 9. COMES HERE FOR END OF SUB TRAJECTORY.
C      IF (DONE) 113,111,111
111     DONE = 0.
      IF (NSTAGE-LSTAGE) 112,115,115
112     NSTAGE = NSTAGE+1
      GO TO 100
113     DONE = 0.
115     CALL EXTRA
      IF (RETURN) 103,116,100
116     RETURN
      END
C
C      SUBROUTINE STEP TESTS FOR THE END OF THE PROBLEM, COMPUTES STEP SIZE, AND
C      CONTROLS QUANTITY OF OUTPUT DATA.  END OF PROBLEM OCCURS IF TIME = TMAX,
C      STEPGO+STEPNO = STEPMX, OR C(LOOKX) = XLOOK.  THE LAST OPTION ALLOWS STOP-
C      PING ON A DEPENDENT VARIABLE.  THE TEST FOR STOPPING AT XLOOK IS NOT MADE
C      UNTIL C(LOOKSW) IS GREATER THAN SWLOOK.  CONTROL ON QUANTITY OF OUTPUT IS
C
C      MODOUT=1  OUTPUT EVERY NTH STEP(N=STEPS) UNTIL TIME = TMIN, THEN
C      GO TO MODE 2
C      2  OUTPUT AT INTERVALS OF DELMAX UNTIL TIME = TMAX.
C      3  OUTPUT AT INTERVALS OF DELMAX UNTIL TIME = TMIN, THEN
C      GO TO MODE 4.
C      4  OUTPUT EVERY NTH STEP UNTIL TIME = TMAX.
C
C      COMMON C
C      DIMENSION A(600), B(700), C(4000),
C      1  XPRIM(200), DELT1 (10)
C
C      EQUIVALENCE
C      1(A ,C ( 11)),(A1 ,B ( 10)),(A2 ,B ( 11)),
C      2(B ,C (111)),(DELMAX,A ( 19)),(DEL ,A ( 43)),
C      3(DELT ,B ( 11)),(DONE ,B ( 39)),(E2 ,B ( 18)),
C      4(END ,A ( 5)),(ERLOG ,B ( 17)),(H2 ,B ( 15)),
C      5(INLOOK,A ( 59)),(LOOKSW,A ( 9)),(LOOKX ,A ( 8)),
C      6(MODOUT,A ( 20)),(NSTAGE,A ( 3)),(DELT1 ,A ( 133)),
C      7(RATIO ,B ( 58)),(SIGNAL,B ( 31)),(SPACES,B ( 16)),
C      8(STEPGO,A ( 41)),(STEPMX,A ( 16)),(STEPNO,A ( 42)),
C      9(STEPS ,A ( 17)),(SWLOOK,A ( 10)),(TABLE ,C (1911))
C      EQUIVALENCE
C      1(TMAX ,B ( 4)),(TMIN ,A ( 18)),(TTOL ,A ( 45)),
C      2(XLOOK ,A ( 12)),(XPRIM ,C ( 711)),(XTOL ,A ( 11)),
C      3(INSTART,B ( 24)),(SWITCH,A (601)),(OUTPUT,B ( 399))
C      CHECKF(A,B,C) = ABSF(A-B) - ABSF(A-C)
C
C      PART 1. TEST FOR END OF THE PROBLEM (MAXIMUM PROBLEM TIME OR MAXIMUM
C      NUMBER OF STEPS).
C      STEPGO = STEPGO + 1.
      OUT = OUTPUT
      IF (ABSF(TMAX-XPRIM(1))-TTOL) 1,1,3
1      DONE = 1.0
112     CALL OUTPUT
      IF (OUTPUT) 26,111,26
111     WRITE OUTPUT TAPE 6,2,NSTAGE
2      FORMAT(6HSTAGE12,11H COMPLETED.//)
      GO TO 26
3      IF (STEPGO+STEPNO-STEPMX) 7,4,4
4      CALL OUTPUT
      WRITE OUTPUT TAPE 6,5,STEPMX
5      FORMAT (22HSTEPGO+STEPNO=STEPMX=F6.)
      CALL EXIT
C

```

```

C   PART 2.  COMPUTE STEP SIZE (DELTA) AND CONTROL OUTPUT.
7   N=1
   A3 = (A2-A1)*RATIO+A2
   AA = (ERLOG-A3)/5.
   IF ((ABSF(AA)-88.028)*ABSF(SWITCH)) 8,8,60
8   DELT = SIGNF(EXPF(AA),DELT)
   IF (DELT/H2-3.) 10,10,9
9   DELT = 3.*H2
10  MODOUT = MODOUT
   GO TO (11,15,13,21),MODOUT
11  IF(DELT*(XPRIM(1) + 3.*DELT-TMIN)) 21,12,12
12  MODOUT = 2
   DEL = TMIN - XPRIM(1)
   GO TO 16
13  IF(DELT * (XPRIM(1) - TMIN)) 15,15,14
14  MODOUT = 4
   GO TO 21
15  DEL = DEL-H2
16  SPACES = INIF((DEL/DELT)+SIGNF(.9,(DEL/DELT)))
17  IF(SPACES) 20, 18,20
18  CALL OUTPUT
   N=2
   DEL = DELMAX
   IF (ABSF(DEL) - ABSF(DELT)) 19,16,16
19  DELT = SIGNF(DEL,DELT)
   GO TO 16
20  DELT = DEL/SPACES
   GO TO 23
21  IF (MODF(STEPCO,STEPS)) 23,22,23
22  CALL OUTPUT
   N=2
C
C   PART. 3.  SEARCH FOR C(LOOKX) = XLOOK UNLESS LOOKX=0.
23  IF(LOOK X) 27,42,27
27  LOOK X = LOOK X
   LOOK SW = LOOK SW
   OUTPOT = 1.
   GO TO (44,45),N
44  CALL OUTPUT
45  IF(SWITCH) 32,28,33
28  IF(SW LOOK - C(LOOK SW)) 29,29,42
29  XTOL1 = XTOL*ABSF(XLOOK)
   IF (XTOL1) 31,30,31
30  XTOL1 = XTOL
31  SWITCH = -1.
   GO TO 41
32  SWITCH = 1.
   ASSIGN 43 TO MODE
   OVER = 0.
   F = 0.
   T=0.
33  SLOPE = (C(LOOKX)-OLDX)/H2
   GO TO MODE, (43,35)
43  IF(SLOPE * (C(LOOK X) - X LOOK)) 350,41,41
35  ASSIGN 35 TO MODE
35  IF(ABSF(C(LOOK X)- X LOOK) - XTOL1) 36,36,37
60  T=1.
36  IF (OUT) 63,46,63
46  OUTPOT = 0.
   CALL OUTPUT
63  IF (T) 61,47,61
61  IF (OUT) 62,51,62
51  WRITE OUTPUT TAPE 6,64, LOOKX,C(LOOKX),H2,LOOKX,SLOPE
64  FORMAT(3HOC(14,4H) = 1PG15.8,31H CONVERGENCE TROUBLE.   DELT=
   1G15.8,14H SLOPE OF C(14,13H) VS. TIME = G15.8//)
   GO TO 62
47  IF (OUT) 62,50,62
50  WRITE OUTPUT TAPE 6,48,LOOK X, C(LOOK X)
48  FORMAT(3HOC(14,2H)=1PG15.8//)
62  LOOKX = 0
   XTOL1 = 0.
   SIGNAL = 1.
   SWITCH = 0.
   DONE = END
   NSTART = 0
   NSTAGE=NSTAGE
   DELT = DELT/(NSTAGE)
49  CALL INPUT(INLOOK,C,TAPE)
   IF (DONE) 110,42,110
110  IF (OUT) 26,111,26
37  SIGN = CHECKF(OLDX,XLOOK,C(LOOK X))
   IF(SIGN) 40,40,38
40  OVER = 1.
   GO TO 400
38  IF (OVER) 400,401,400
401  XGUESS = C(LOOKX)+SLOPE*DELT
   IF (CHECKF(C(LOOKX),XLOOK,XGUESS)) 402,41,41
402  F = F+1.
   IF (F-7.) 400,400,403
403  SLOPE = SLOPE/F
400  IF (SLOPE) 404,60,404
404  DELT = SIGNF(ABSF(XLOOK-C(LOOKX))/SLOPE,SIGN*H2)
41  OLDX = C(LOOK X)
42  IF (ABSF(TMAX-XPRIM(1))-ABSF(DELT)) 25,26,26
25  DELT = TMAX-XPRIM(1)
   GO TO (26,24,24,26),MODOUT
24  DEL = DEL-DELT
26  OUTPOT = OUT
   RETURN
   END

```

```

C                                     SUBROUTINE STDATA
C THIS ROUTINE CLEARS THE A, XPRIM, XPRIMB ARRAYS AND LOADS A SET OF
C STANDARD DATA INTO THE MACHINE. ANY VALUES SET HERE MAY BE OVERWRITTEN BY
C INPUT 1 IN THE MAIN PROGRAM.
C
C COMMON C
C
C   DIMENSION A(600), B(700), C(4000),
C   1 PNAME (12),          AMASS (30),          XPRIM (200),
C   2                      COEFN (190),          ICC (6),
C   3 AK (3),              XDOT (100),          IND (3),
C   4 REFER (12),          RCRIT (30),          AW (4),
C   5 RMASS1 (10)
C
C   EQUIVALENCE
C   1(A ,C ( 11)),(AK ,A ( 51)),(AMASS ,A ( 347)),
C   2(AU ,A ( 29)),(AW ,A ( 55)),(B ,C (111)),
C   3(BODYCD,A ( 143)),(COEFN ,A ( 407)),(CONSTU,A ( 32)),
C   4(CONSU ,A ( 31)),(DIOFFJ,A ( 23)),(EREF ,A ( 13)),
C   5(ERLIMT,A ( 14)),(ETOL ,A ( 30)),(GASFAC,A ( 46)),
C   6(ICC ,A ( 153)),(IMODE ,A ( 11)),(IND ,A ( 60)),
C   7(LOOKSW,A ( 9)),(MUDOUT,A ( 20)),(NEQ ,A ( 2)),
C   8(NSTAGE,A ( 3)),(OBLATD,A ( 27)),(OBLATH,A ( 28)),
C   9(OBLATJ,A ( 26)),(PNAME ,A ( 287)),(RCRIT ,A ( 377))
C   EQUIVALENCE
C   1(REFER ,A ( 317)),(RE ,A ( 25)),(SPD ,A ( 44)),
C   2(SQRDK1,A ( 47)),(STEPMX,A ( 16)),(STEPS ,A ( 17)),
C   3(TFILE ,A ( 6)),(XDOT ,B ( 501)),(XPRIM ,C ( 711)),
C   4(XTOL ,A ( 11)),(RMASS1,A ( 73))
C
C   CLEAR INITIAL CONDITIONS AND CONTROL PARAMETERS.
C   DO 1 J=1,1100
C   1 A(J) = 0.
C
C   THE FOLLOWING NH STATEMENTS LOAD THE BODY NAMES INTO THE MACHINE.
C   PNAME(1) = 3HSUN
C   PNAME(2) = 6HMERCUR
C   PNAME(3) = 5HVENUS
C   PNAME(4) = 5HEARTH
C   PNAME(5) = 4HMARS
C   PNAME(6) = 6HJUPITE
C   PNAME(7) = 6HSATURN
C   PNAME(8) = 6HURANUS
C   PNAME(9) = 6HNEPTUN
C   PNAME(10) = 5HPLUTO
C   PNAME(11) = 4HMOON
C   PNAME(12) = 6HEARTHM
C
C   FILL OUT SUN REFERENCE LIST. INITIALIZE MASS ARRAY.
C   DO 2 K=1,10
C   RMASS1(K) = 1.
C   2 REFER(K+1) = PNAME(1)
C   REFER(12) = PNAME(1)
C
C   FILL OUT EARTH REFERENCE LIST.
C   REFER(1) = PNAME(4)
C   REFER(4) = 5HZERO+
C   REFER(11) = PNAME(4)
C

```

```

C      LOAD THE REMAINING STANDARD DATA.
      AK(1) = 0.5
      AK(2) = 0.5
      AK(3) = 1.0
      AMASS(1) = 1.0
      AMASS(2) = 1.0/6120000.0
      AMASS(3) = 1.0/408645.0
      AMASS(4) = 1.0/332951.3
      AMASS(5) = 1.0/3088000.0
      AMASS(6) = 1.0/1047.39
      AMASS(7) = 1.0/3500.0
      AMASS(8) = 1.0/22869.0
      AMASS(9) = 1.0/18889.0
      AMASS(10) = 1.0/400000.0
      AMASS(11) = AMASS(4)/81.335
      AMASS(12) = AMASS(4) + AMASS(11)
      AU = 1.49599 E11
      AW(1) = 1./6.
      AW(2) = AW(1) + AW(1)
      AW(4) = AW(1)
      AW(3) = 1. - (AW(2) + (AW(1) + AW(4)))
      BODYCD = PNAME(4)
      COEFN(1) = -1E20
      COEFN(189) = 1E20
      CONSTU = 1.0 E-6
      CONSU = 1E-6
      ETOL = 0.01
      DTIOFFJ = 244.E4
      EREF = 1E-6
      ERLIMT = 3E-6
      GASFAC = 20.064881
      ICC(1) = 185
      ICC(2) = 185
      ICC(3) = 185
      ICC(4) = 185
      IMODE = 1
      IND(1) = 2
      IND(2) = 3
      IND(3) = 1
      LOOKSW = 711
      MODDUT = 4
      NEQ = 8
      NSTAGE = 1
      OBLATJ = 1.62345 E-3
      OBLATH = -5.75 E-6
      OBLATD = 7.875 E-6
      RCRIT(1) = 1.0 E+20
      RCRIT(2) = 1.0 E+8
      RCRIT(3) = 6.14 E+8
      RCRIT(4) = 9.25 E+8
      RCRIT(5) = 5.78 E+8
      RCRIT(6) = 4.81 E+10
      RCRIT(7) = 5.46 E+10
      RCRIT(8) = 5.17 E+10
      RCRIT(9) = 8.61 E+10
      RCRIT(10) = 3.81 E+10
      RCRIT(11) = 1.60 E+8
      RE = 6378165.
      SPD = 86400.0
      SQROK1 = 2.959122083 E-4
      STEPMX = 100.0
      STEPS = 1.
      TFILE = 1.
      XDOT(1) = 1.0
      XPRIM(2) = KMASS1(1)
      XTOL = 5E-8
      WRITE OUTPUT TAPE 6,3
3  FORMAT (15H0STANDARD DATA.)
      RETURN
      END

```

```

C SUBROUTINE TESTTR MAY BE CALLED FOR ONE OF TWO REASONS, (1) TO TEST FOR AND
C POSSIBLY TRANSLATE THE ORIGIN (WHEN IMODE IS +) OR (2) TO CHANGE THE
C VARIABLES OF INTEGRATION (WHEN IMODE IS -). A TRANSLATION OF THE ORIGIN
C OCCURS WHEN THE OBJECT MOVES INTO A SPHERE OF INFLUENCE WHICH IS SMALLER
C THAN ANY OTHERS IT MAY ALSO BE IN. WHEN THIS HAPPENS, THE NAME OF THE NEW
C ORIGIN IS MOVED TO THE BEGINNING OF THE BNAME LIST AND ORDER IS
C CALLED TO REORDER THE BNAME LIST.
C
C COMMON C
C
C DIMENSION A(600), B(700), C(4000),
C 1 XPRIM(100,2),XPRIMB(100,2),XWHOLE(6),VEFM(3,8),VX(3),
C 2ORBELS(6),BMASS(8),BNAME(8),RB(3,8),RBCRIT(8),R(8)
C
C EQUIVALENCE
C 1(A,C ( 11)),(AMC,B ( 87)),(ASYMPT,A ( 7)),
C 2(B,C (111)),(BMASS,B ( 137)),(BNAME,B ( 122)),
C 3(CHAMP,B ( 25)),(DELTA,B ( 1)),(GK2M,B ( 36)),
C 4(IMODE,A ( 1)),(NBODYS,B ( 41)),(ORBELS,B ( 116)),
C 5(RBCRIT,B ( 145)),(RB,B ( 193)),(REVS,A ( 48)),
C 6(R,B ( 102)),(SQDK,B ( 35)),(TABLE,C (1911)),
C 7(TMAX,B ( 4)),(TRANSFER,B ( 8)),(TRU,B ( 40)),
C 8(TTEST,A ( 54)),(VEFM,B ( 241)),(VX,B ( 92)),
C 9(XPRIM,C ( 711)),(XPRIMB,C ( 911)),(XWHOLE,B ( 110))
C EQUIVALENCE
C 1(OUTPUT,B ( 399))
C
C IMODE = IMODE
C IF (IMODE) 12,12,1
C
C IF IMODE IS +, TEST FOR TRANSLATION OF THE ORIGIN.
C
C 1 CHAMP= 1.E+30
C DO 4 JB=1,NBODYS
C IF (R(JB)-RBCRIT(JB)) 2,4,4
C 2 IF (CHAMP-RBCRIT(JB)) 4,4,3
C 3 CHAMP = RBCRIT(JB)
C NCHAMP = JB
C 4 CONTINUE
C IF (NCHAMP-1) 26,26,5
C 5 TRANSFER = 1.0
C 8 BTEMP = BNAME(1)
C BNAME(1) = BNAME(NCHAMP)
C BNAME(NCHAMP) = BTEMP
C TTEST = 0.
C REVS = 0.
C IF (OUTPUT) 6,9,6
C 9 WRITE OUTPUT TAPE 6,10,BNAME(NCHAMP),BNAME(1)
C 10 FORMAT (28HOORIGIN IS TRANSLATING FROM A6,4H TO A6)
C 6 CALL EPHMRS
C DO 11 K=1,3
C VX(K) = VX(K)-VEFM(K,NCHAMP)
C RB(K) = RB(K,NCHAMP)
C XPRIM(K+2,1)=VX(K)
C XPRIM(K+5,1)=RB(K)
C XPRIMB(K+2,1) = 0.
C XPRIMB(K+5,1) = 0.
C XWHOLE(K)= VX(K)
C 11 XWHOLE(K+3) = RB(K)
C GO TO 20
C
C IF IMODE IS -, CHANGE THE VARIABLES OF INTEGRATION.
C
C 12 DO 13 K=1,3
C XPRIM(K+2,1)=XWHOLE(K)
C XPRIM(K+5,1)=XWHOLE(K+3)
C XPRIMB(K+2,1) = 0.
C XPRIMB(K+5,1) = 0.
C VX(K) = XWHOLE(K)
C 13 RB(K) = XWHOLE(K+3)
C GO TO (16,14,15),IMODE
C 14 CODE = 5HORBIT
C IMODE = 1
C GO TO 18
C 15 IMODE = 3
C GO TO 17
C 16 IMODE = 2
C 17 CODE = 6HRECTAN
C 18 NCHAMP = 1
C IF (OUTPUT) 20,7,20
C 7 WRITE OUTPUT TAPE 6,19,CODE
C 19 FORMAT (33HOINTEGRATION MODE IS CHANGING TO A6)
C 20 GO TO (21,26,26),IMODE
C 21 CALL CONV1(VX,AMC)
C GK2M= SQDK*(BMASS(NCHAMP)+XPRIM(2,1)/1.9866 E+30)
C 30 CALL CONV2
C IF ORIGIN TRANSLATION CAUSES PATH TO LIE NEAR AN ASYMPTOTE, CHANGE
C INTEGRATION VARIABLES TO RECTANGULAR IF THEY ARE ORBIT ELEMENTS.
C IF (ORBELS(1)-1.) 24,24,22
C 22 IF (ABS( (TRU)-2.3/SQRT(ORBELS(1))) 24,24,23
C 23 ASYMPT = 1.0
C GO TO 15
C 24 DO 25 J=1,6
C 25 XPRIM(J+2,1) = ORBELS(J)
C 26 IF (TRANSFER) 27,28,27
C 27 CALL INPUT (101,C, TABLE)
C 29 CALL ORDER
C 28 RETURN
C END

```

```

C                                     SUBROUTINE THRUST
C THIS ROUTINE COMPUTES X,Y,Z THRUST ACCELERATIONS. THE THRUST VECTOR IS
C ASSUMED COINCIDENT WITH THE LONGITUDINAL AXIS OF THE VEHICLE, WHICH IS
C ORIENTED TO THE RELATIVE WIND VELOCITY BY THE ANGLE OF ATTACK (ALPHA) AND
C THE ROLL ANGLE (BETA). ALPHA IS ASSUMED TO BE A QUADRATIC FUNCTION OF TIME
C WHEREAS BETA IS ASSUMED TO BE CONSTANT.
C REVOLV IS THE EARTHS ROTATION RATE IN RADIANS/SEC (7.29211585E-5) AND THE
C FACTOR 8589934592.= 2**33 IS REMOVED TO PREVENT OVERFLOW.
C
C COMMON C
C
C DIMENSION A(600), B(700), C(4000),
C 1 FORCE(3), PAR(3), VATM(3), P(3), IND(3),RAMC(5),RB(3),X(100)
C
C EQUIVALENCE
C 1(A ,C ( 11)),(AEXIT ,B ( 3)),(ALPHA ,A ( 49)),
C 2(B ,C (111)),(BETA ,A ( 50)),(COSALF,B ( 48)),
C 3(COSBET,B ( 49)),(EXITA ,B ( 39)),(FLOW ,B ( 5)),
C 4(FORCE ,B ( 66)),(IND ,A ( 60)),(PAR ,B ( 60)),
C 5(PMAGN ,B ( 50)),(PRESS ,B ( 33)),(P ,B ( 84)),
C 6(PUSHO ,B ( 39)),(PUSH ,A ( 166)),(RAMC ,B ( 393)),
C 7(RATMOS,B ( 23)),(RB ,B ( 193)),(REVOLV,B ( 21)),
C 8(R ,B ( 102)),(RSQRD ,B ( 45)),(SIMP ,B ( 2)),
C 9(SINALF,B ( 46)),(SINBET,B ( 47)),(VATM ,B ( 97))
C EQUIVALENCE
C 1(VQ ,B ( 100)),(VQSQRD,B ( 101)),(VX ,B ( 92)),
C 2(VY ,B ( 93)),(VZ ,B ( 94)),(X ,B ( 401))
C
C
C SINBET = SIN(BETA/57.2957795)
C COSBET = COS(BETA/57.2957795)
C VATM(1)=VX*REVOLV*RB(2)
C VATM(2)=VY*REVOLV*RB(1)
C VATM(3)=VZ
C 3 CALL CONVT1(VATM,RAMC)
C 4 ALPHA = QUAD(X(1),1)/57.2957795
C SINALF=SIN(ALPHA)
C COSALF=COS(ALPHA)
C DO 1 J1=1,3
C J2=IND(J1)
C J3=IND(J2)
C 1 P(J1) = (VATM(J2)*RAMC(J3)-VATM(J3)*RAMC(J2))/8589934592.
C PMAGN= SQRT(P(1)*P(1)+P(2)*P(2)+P(3)*P(3))
C PUSH = PUSHO-EXITA*PRESS
C TDPMAG = PUSH/PMAGN/X(2)
C R4 = SINBET/VQ
C R5 = COSALF/RAMC(4)
C DO 2 J1=1,3
C J2=IND(J1)
C J3=IND(J2)
C PAR(J1)=P(J2)*VATM(J3)-P(J3)*VATM(J2)
C 2 FORCE(J1) = TDPMAG*(SINALF*(COSBET*P(J1)+R4*PAR(J1))-R5*(P(J2)*
C 1 RAMC(J3)-P(J3)*RAMC(J2)))
C RETURN
C END
C
C                                     SUBROUTINE TUDES
C THIS ROUTINE COMPUTES THE RECTANGULAR POSITION AND VELOCITY COMPONENTS
C WITH RESPECT TO THE EARTH MEAN EQUINOX AND EQUATOR OF 1950.0 FROM THE
C LATITUDE, LONGITUDE, AZIMUTH, ELEVATION, ALTITUDE, TOTAL VELOCITY, AND
C TIME. ALSO, WHEN TKICK DOES NOT EQUAL ZERO, A NON-DRAG VERTICAL STEP OF
C SIZE TKICK IS MADE IN CLOSED FORM (STATEMENTS 2 TO 4). THE INTEGRATION
C WILL THEN BEGIN AT TIME EQUAL TO TIME+TKICK WITH THE ORIENTATION SPECIFIED
C BY THE ABOVE FOUR ANGLES AND THE COMPUTED VALUES OF ALTITUDE AND VELOCITY.
C FOR THE CLOSED FORM APPROXIMATION, A CONSTANT FLOW RATE (FLOW), VACUUM
C SPECIFIC IMPULSE (SIMP) AND ENGINE EXIT AREA (AEXIT) ARE ASSUMED KNOWN.
C THE ATMOSPHERIC PRESSURE IS TAKEN TO BE THE SEA LEVEL VALUE.
C
C COMMON C
C
C DIMENSION A(600), B(700), C(4000),
C 1 SINA(4), COSA(4), ANGLEB(4), XPRIM(200)
C
C EQUIVALENCE
C 1(A ,C ( 11)),(AEXIT ,B ( 3)),(ALT ,A ( 4)),
C 2(AZI ,A ( 35)),(B ,C (111)),(DTOFFJ,A ( 23)),
C 3(ELEV ,A ( 36)),(FLOW ,B ( 5)),(GK2M ,B ( 36)),
C 4(LAT ,A ( 33)),(LONG ,A ( 34)),(OBLATJ,A ( 26)),
C 5(OBLATN,A ( 40)),(RE ,A ( 25)),(RESQRD,B ( 7)),
C 6(ROTATE,A ( 39)),(SIMP ,B ( 2)),(SPD ,A ( 44)),
C 7(STEPGO,A ( 41)),(STEPNO,A ( 42)),(TKICK ,A ( 15)),
C 8(TOFFT ,A ( 24)),(VEL ,A ( 37)),(XPRIM ,C ( 711)),
C 9(OUTPOT,B ( 399))
C EQUIVALENCE (QLAT,LAT),(QLONG,LONG)
C

```

```

      ALT1 = 0.
      VEL1 = VEL
      DEL1 = 0.
      DEL = 0.
      ASSIGN 1 TO NGO
      DAYS = DTOFFJ - 2433282.5
      GREEN = MOUF(100.0755426+.985647346DAYS+2.9015E-13DAYS**2
1+7.29211585E-5*(TOUFFT*SPU*XPRIM(1))*57.2957795,360.)
      SINA(1) = SIN(QLAT/57.2957795)
      IF (OBLATN) 102,101,102
101 RADIUS = RE + ALT
      GO TO 8
102 RADIUS=6356783.28/SQRTF(1.9933065783+.006693421685*SINA(1)**2)+ALT
      GO TO 8
      1 XPRIM(6) = COSA(2)*COSA(1)*RADIUS
      XPRIM(7) = SINA(2)*COSA(1)*RADIUS
      XPRIM(8) = SINA(1)*RADIUS
      RMASSO = XPRIM(2)
      XPRIM(2) = XPRIM(2)-FLOW*TKICK
      IF (OUTPOT) 12,11,12
11 WRITE OUTPUT TAPE 6,3,STEPGO,STEPNO,LAT, LONG,AZI,ELEV,ALT,XPRIM
11),VEL,RMASSO,(XPRIM(J),J=6,8)
3 FORMAT(6HSTEP=F5.2H +F4.4X,6H LAT.=1PG15.8,7H LONG.=G15.8,6H AZ
11.=G15.8,7H ELEV.=G15.8,6H ALT.=G15.8/6H TIME=G15.8,6H VEL.=G15.8,
6H RMASS=G15.8,4X,2HX=G15.8,5X,2HY=G15.8,4X,2HZ=G15.8)
12 IF (TKICK) 2,50,2
2 XPRIM(1) = XPRIM(1)+TKICK
B1 = LOGF(RMASSO/XPRIM(2))
SIMPSL = SIMP-AEXIT/FLOW*10332.275
VEL1 = VEL+SIMPSL*9.80665*81-G*TKICK
ALT1 = TKICK*(VEL-G*TKICK/2.+9.80665*SIMPSL*(1.-B1*XPRIM(2)/
1 (RMASSO-XPRIM(2))))
4 RADIUS = RADIUS + ALT1
GREEN = GREEN + 7.29211585E-5*TKICK*57.2957795
ASSIGN 5 TO NGO
GO TO 8
5 XPRIM(6) = COSA(2)*COSA(1)*RADIUS
XPRIM(7) = SINA(2)*COSA(1)*RADIUS
XPRIM(8) = SINA(1)*RADIUS
50 IF (OBLATN) 6,7,6
6 DEL1 = ATANF((C2-L.)/(C3-L.)*SINA(1)/COSA(1))*57.2957795-QLAT
7 DEL2 = RADIUS/G*SINA(1)*COSA(1)*ROTATE*ROTATE*57.2957795L
DEL = DEL1 + DEL2
ASSIGN 10 TO NGO
8 ANGLEB(1) = QLAT + DEL
ANGLEB(2) = QLONG + GREEN
ANGLEB(3) = AZI
ANGLEB(4) = ELEV
DO 9 I=1,4
SINA(I) = SIN(ANGLEB(I)/57.2957795)
9 COSA(1) = COSF(ANGLEB(1)/57.2957795)
C1 = 5.*RESQRD/RADIUS/RADIUS*OBLATJ
C2 = C1*(SINA(1)*SINA(1)-.6)
C3 = C1*(SINA(1)*SINA(1)-.2)
G = GK2M/RADIUS/RADIUS
GO TO NGO, (1,5,10)
10 COS1 = COSA(1)*SINA(4)-COSA(4)*COSA(3)*SINA(1)
COS2 = COSA(4)*SINA(3)
XPRIM(3) = VEL1*(COS1*COSA(2)-COS2*SINA(2))-XPRIM(7)*ROTATE
XPRIM(4) = VEL1*(COS1*SINA(2)+COS2*COSA(2))+XPRIM(6)*ROTATE
XPRIM(5) = VEL1*(SINA(1)*SINA(4)+COSA(1)*COSA(3)*COSA(4))
RETURN
END

      SUBROUTINE TAPE
C      SUBROUTINE TAPE USES THE MASTER MERGED EPHEMERIDES TAPE (TAPE 9 AT LEWIS)
C      TO COMPILE A WORKING EPHEMERIS TAPE (TAPE 3 AT LEWIS) WHICH CONTAINS ONLY
C      THAT DATA NEEDED AT EXECUTION TIME. THIS MINIMIZES TAPE HANDLING DURING
C      EXECUTION. 2 EPHEMERIS FILES ARE ON TAPE 9, FIRST FILE HAS DATA AND IS
C      IDENTIFIED BY THE SECOND WORD OF EACH 256 WORD RECORD (FIRST WORD IS THE
C      DUMMY FORTRAN COMPATIBLE WORD, SECOND WORD=2). THE SECOND FILE IS ONLY 2
C      WORDS LONG, FIRST WORD IS FORTRAN COMPATIBLE, SECOND WORD=3).
C      MASTER FILE 1 -- PLANETS (EXCEPT MERCURY AND EARTH), SUN, MOON, AND
C      EARTH-MOON BARYCENTER FROM SEPT.25, 1960 TO ABOUT 2000.
C      EACH EPHEMERIS COMPILED REQUIRES A SET OF INPUT 300 DATA. THE FIRST PIECE
C      OF DATA WRITTEN ON A FILE IS THE FILE IDENTIFICATION NUMBER, FILE. EACH
C      FILE IS NUMBERED CONSECUTIVELY STARTING WITH FILE=1. SINCE MOON DATA IS IN
C      TERMS OF EARTH RADII, THE CONVERSION OF MOON DATA TO A.U. IS MADE BEFORE
C      WRITING ON TAPE 3. THE COMMON USED IN SUBROUTINE TAPE IS LOCAL AND ALL
C      BUT TAPE3 IS CLEARED BY A FINAL CLEARING LOOP.
C      FUNCTION COMPARE(A,B) IS EQUIVALENT TO (A-B) BUT WILL NOT OVERFLOW.
C      NORMAL INPUT - ELIST, TBEGIN, TEND, TAPE3
C
C      ELIST- THE BCD LIST OF EPHEMERIS DATA NAMES TO BE PLACED ON
C      TAPE 3. THE NAMES ARE READ FROM CARDS, AND IS USED TO
C      MAKE THE TMAKE LIST. ELIST IS NOT CHANGED IN STORAGE UNTIL
C      THE FINAL CLEAR FOR THIS SUBROUTINE.
C      TMAKE- THE LIST OF EPHEMERIS NAMES WITH DUPLICATES DROPPED AND
C      ZERO SPACES CLOSED IN. AS THE EPHEMERIDES ARE FINISHED THE
C      NAMES ARE ERASED FROM THIS LIST.
C      TMADE- LIKE TMAKE BUT IS HELD FOR OUTPUT.
C      TBEGIN- THE BEGINNING DATE EXPRESSED AS A JULIAN DAY.
C      TEND- ENDING DATE EXPRESSED AS A JULIAN DAY.
C      INTVAL- THE APPROX. NUMBER OF DAYS COVERED BY ONE SET OF COEFF. IT
C      IS USED TO DECIDE WHICH DATA ARE TO BE ENTERED DOUBLE. THE
C      DOUBLE ENTRIES PERMIT FASTER OPERATION IF REVERSAL OF
C      INTEGRATION IS REQUIRED FOR ANY REASON.
C      EDATE- JULIAN ENDING DATE FOR THE MASTER EPHEMERIS.
C      ERTAU- EARTH RADII PER A.U.
C
C      COMMON C

```

```

      DIMENSION
      1      C (700),      TMAKE (12),      LIST (30),
      2      EDATE (12),      INTVAL (30),      KTAG (12),
      3      ELIST (11),      TMADE (12),      INTVA (2),
      4      PNAME (30),      TDATUM (252),      DATUMT (21,12)
C
      EQUIVALENCE
      1( TAPE3,C( 2)),(ERTOAU,C( 3)),( KTAG,C( 4)),( FILE,C(16)),
      2( ELIST,C(17)),(TBEGIN,C(29)),( TEND,C(30)),( PNAME,C(31)),
      3( KHAMP,C(61)),( TMADE,C(73)),( TMAKE,C(85)),(TDATUM,C(441)),
      4( EDATE,C(127)),(INTVAL,C(157)),( INTVA,C(156)),(DATUMT,C(189))
C
      COMPARE(A,B) = (A+B)*(-(A*B))
      REMIND 3
      DO 1 K=1,4000
1  C(K) = 0.0
C
      THE FOLLOWING NH STATEMENTS LOAD THE BODY NAMES INTO THE MACHINE.
C
      NOTE. THE EARTH IS NOT IN THIS LIST (NO EPHEMERIS FOR EARTH.)
      PNAME(1) = 3HSUN
      PNAME(2) = 6HMERCUR
      PNAME(3) = 5HVENUS
      PNAME(4) = 4HMARS
      PNAME(5) = 6HJUPITE
      PNAME(6) = 6HSATURN
      PNAME(7) = 6HURANUS
      PNAME(8) = 6HNEPTUN
      PNAME(9) = 5HPLUTO
      PNAME(10) = 4HMOON
      PNAME(11) = 6HEARTH
C
      PART 2. SET UP JULIAN DATES ENDING EACH EPHEMERIS.
C
      EDATE(1) = 2451872.5
      EDATE(3) = 2451848.5
      EDATE(4) = 2451020.5
      EDATE(5) = 2473520.5
      EDATE(6) = 2473520.5
      EDATE(7) = 2473520.5
      EDATE(8) = 2473520.5
      EDATE(9) = 2473520.5
      EDATE(10) = 2440916.5
      EDATE(11) = 2451848.5
      INTVA = 30000
      INTVAL(1) = 8
      INTVAL(2) = 5
      INTVAL(3) = 15
      INTVAL(4) = 44
      INTVAL(5) = 330
      INTVAL(6) = 825
      INTVAL(7) = 1211
      INTVAL(8) = 1172
      INTVAL(9) = 1101
      INTVAL(10) = 2
      INTVAL(11) = 15
      FILE = 1.
      ERT0AU = 4.26546512 E-5
      MOON = 0
      LI = 1
C
      PART 2B. CALL INPUT AND SEE IF TAPE IS TO BE MADE. INPUT MUST ALWAYS
      MAKE TAPE3=0.0 IF TAPE IS TO BE MADE.
C
      TAPE3 = 3.
      8 CALL INPUT(300,C,LIST)
      IF (TAPE3) 63,3,63
      3 IF (FILE-1.) 20,10,20
      10 CALL SKFILE(9,2)
C
      PART 3. TAPE IS TO BE MADE SO MOVE EPHEMERIS LIST TO TMAKE AND
      TO TMADE (FOR OUTPUT), CANCEL ANY ZERO OR DUPLICATE NAMES.
C
      20 KOUNT = 1
      DO 6 K=1,11
      TMAKE(K) = 0.
      TMADE(K) = 0.
      4 DO 5 J=1,KOUNT
      IF (COMPARE(ELIST(K),TMAKE(J-1))) 5,6,5
      5 CONTINUE
      TMAKE(KOUNT) = ELIST(K)
      TMADE(KOUNT) = ELIST(K)
      KOUNT = KOUNT+1
      6 CONTINUE
      KOUNT = KOUNT - 1
C
      PART 4. FIND INPUT ERRORS.
C
      7 IF(TBEGIN-2437202.5) 66,9,9
      9 KM = 2
      11 ERROR = 0.
      WRITE TAPE 3,FILE
      DO 21 J=1,KOUNT
      KTAG(J) = 0
      12 DO 13 K=1,20
      IF (COMPARE(PNAME(K),TMAKE(J))) 13,16,13
      13 CONTINUE
C
      PART 5. PRINTS OUT THE MISPELLED NAMES AND OTHER ERRORS.
C
      14 PRINT 15, TMAKE(J), TBEGIN, TEND
      WRITE OUTPUT TAPE 6, 15, TMAKE(J), TBEGIN, TEND, (PNAME(K),
      1 EDATE(K),K=1,20)
      15 FORMAT(1,23H TROUBLE ON TAPE 3 MAKE / 2X,A6,10H T BEGIN= F10.1,8H
      1 T END= F10.1//2(2X,A6,F20.1))
      ERROR = 1.
      GO TO 21
C

```



```

C   PART 4B. CHECKS DATES AND STORES INDEX FOR MOON SO THAT EARTH
C   RADII CAN BE CONVERTED TO A.U.
16 IF (10-K) 18,17,18
17 MOON = J
18 KTAG(J) = K
19 IF (EDATE(K)- TEND) 14,21,21
21 CONTINUE
   ASSIGN 36 TO NS1
   IF (ERROR) 22,22,68
C
C   PART 6. FIX UP A TAG (KTAG) TO INDICATE WHETHER TO ENTER DATA DOUBLE OR
C   NOT. KHAMP WILL BE SHORTEST INTERVAL. KTAG WILL BE NON-ZERO IF
C   ANY DATA ENTERS MORE THAN ONCE FOR 10 ENTRIES OF THE MOST
C   FREQUENT DATA.
22 KHAMP = INTVAL(0)
   DO 23 J=1,KOUNT
   K = KTAG(J)
   KHAMP = XMINOF(KHAMP,INTVAL(K))
23 CONTINUE
   KHAMP = KHAMP *10
   DO 24 J=1,KOUNT
   K = KTAG(J)
24 KTAG(J) = INTVAL(K) / KHAMP
C
C   PART 7. LOCATE FILE 2 ON TAPE 9.
25 READ TAPE 9, KFILE
26 IF (KM-KFILE) 27,31,29
27 IF (KFILE - 3) 28,28,29
28 BACKSPACE 9
   BACKSPACE 9
   CALL BSFILE(9)
   GO TO 25
C   BY PASS A FILE.
29 CALL SKFILE(9)
   GO TO 25
C
C   PART 8. THIS IS CORRECT FILE ON TAPE 9. READ DATA. THERE CAN BE UP
C   TO 12 SETS OF DATA PER RECORD. A SET OF DATA IS 21 WORDS.
31 BACKSPACE 9
32 READ TAPE 9, KTAPE,(TDATUM(I), I=1,252)
   GO TO NS1, (36,46)
C
C   PART 9. IS THIS A SATISFACTORY STARTING POINT, QUESTION MARK.
C   THE 1ST SET OF DATA FOR EACH PLANET MUST PRE DATE TBEGIN.
C   PART 9 IS EXECUTED ONLY ONCE.
36 DO 42 J=1,KOUNT
   DO 37 K=1,232,21
   IF (COMPARF(TDATUM(K),TMAKE(J))) 37,39,37
37 CONTINUE
38 LI = J
   BACKSPACE 9
   BACKSPACE 9
   GO TO 32
39 IF (TDATUM(K+1)-TDATUM(K+2)-TBEGIN) 40,40,38
40 DO 41 KJ=1,21
   K1 = K + KJ - 1
41 DATUM(KJ,J) = TDATUM(K1)
42 CONTINUE
   IF (MOON) 43,45,43
43 DO 44 KJ=4,21
44 DATUM(KJ,MOON) = DATUM(KJ,MOON)*ERTOAU
45 ASSIGN 46 TO NS1
C
C   PART 10. PUT AWAY NEEDED DATA. TEST NAME, TIME OF BEGIN AND END. DO NOT
C   WRITE TAPE 3 UNTIL TBEGIN PREDATES THE END OF THE FITTED
C   INTERVAL. 50 REPEATS OLD DATA, 57 WRITES NEW DATA. THE NAMES
C   ARE ERASED FROM TMAKE AS SOON AS THE DATA POST DATES TEND. WHEN
C   ALL NAMES ARE GONE, RETURN TO INPUT 300 TO SEE IF ANOTHER
C   EPHEMERIS IS TO BE CONSTRUCTED.
46 DO 65 K=1,232,21
   DO 47 J=1,KOUNT
   IF (COMPARF(TDATUM(K),TMAKE(J))) 47,48,47
47 CONTINUE
   GO TO 65
48 SWT = TBEGIN-TDATUM(K+1)-TDATUM(K+2)
   IF (SWT) 49,49,52
49 IF(KTAG(J)) 50,52,50
50 WRITE TAPE 3,(DATUM(KJ,J) , KJ=1,21)
52 DO 53 KJ=1,21
   K1 = K + KJ
53 DATUM(KJ,J) = TDATUM(K1-1)
   IF (J-MOON) 56,54,56
54 DO 55 KJ = 4,21
55 DATUM(KJ,J) = DATUM(KJ,J)*ERTOAU
56 IF (SWT) 57,57,58
57 WRITE TAPE 3,(DATUM(KJ,J),KJ=1,21)
58 IF(TEND-DATUM(2,J)-DATUM(3,J)) 59,59,65
59 TMAKE(J) = 0.
   DO 60 KK=1,KOUNT
   IF (TMAKE(KK)) 65,60,65
60 CONTINUE
   WRITE OUTPUT TAPE 6, 61, FILE,TBEGIN,TEND, KOUNT,(TMAKE(KK),
   1KK=1,KOUNT)
61 FORMAT(28H0EPHEMERIS COMPLETED, FILE=F3.,6H, FROM F10.1,3H TO
   1 F10.1, 4H FOR 12, 18H BODIES AS FOLLOWS / 12(2X,A6))
   FILE = FILE + 1.
   END FILE 3
   GO TO 2
63 WRITE TAPE 3, FILE
   REWIND 3
   REWIND 9
   TAPE3 = 3.
   DO 64 J=3,4000
64 C(J) = 0.
   RETURN
C

```

```

65 CONTINUE
GO TO 32
66 PRINT 67, TBEGIN
WRITE OUTPUT TAPE 6,67,TBEGIN
67 FORMAT(33H TBEGIN PREDATES 2437202.5,IT IS F10.1)
68 CONTINUE
REWIND 9
END

      REM BSFILE(I,J) BACKSPACES TAPE I UNTIL IT IS POSITIONED JUST
      REM BEHIND THE J TH EOF MARK.
      REM
      ENTRY   BSFILE
      PZE
      PZE
      PZE
      BCD 1BSFILE
BSFILE SXD   *-4,1
      SXD   *-4,2
      SXD   *-4,4
      XEC*   $(IES)
      TSX   $(RER),4
      LXD   BSFILE-2,4
      CLA*   1,4
      TSX   $(IOS),4
      CLA*   $(RDS)
      STA   BSF
      ANA   A07000
      STA   BTT1
      STA   BTT2
      LXD   BSFILE-2,4
      CAL   2,4
      ANA   =0777777700000
      ERA   =0007400000000
      TNZ   ONEARG
      CLA*   2,4
      TZE   BACK
      PDX   ,1
      AXC   **1,4
      XEC*   $(TCO)
      BTT1  BTTA   **
      TRA   **1
      BSF   BSFA   **
      XEC*   $(RDS)
      XEC*   $(BSR)
      AXC   **1,4
      XEC*   $(TCO)
      BTT2  BTTA   **
      TRA   CHECK
      TIX   BSF,1,1
      XEC*   $(RDS)
      BACK  AXC   **1,4
      XEC*   $(TCO)
      AXC   **1,4
      XEC*   $(TRC)
      NOP
      AXC   **1,4
      XEC*   $(TEF)
      NOP
      LXD   BSFILE-4,1
      LXD   BSFILE-3,2
      LXD   BSFILE-2,4
      TRA   3,4
      CHECK TXL   BACK,1,1
      LXD   BSFILE-2,4
      CLA   ERR+1
      STO   0
      CLA*   1,4
      LDQ*   2,4
      ERR   TSX   8,4
      TXI   BACK,0,14
      PZE   BSFILE-2,0,ERR
      ONEARG CLA   BSFILE-2
      ADD   =01000000
      STO   BSFILE-2
      LXD   CHECK,1
      TRA   BTT1-2
      A07000 OCT 7000
      END

```

```

      REM SKFILE(I,J) SKIPS TAPE I OVER J EOF MARKS.
      REM
      ENTRY   SKFILE
      PZE
      PZE
      PZE
SKFILE  SXD  *-3,1
        SKD  *-3,2
        SXD  *-3,4
        TSX  $(RER),4      CHECK LAST READ
        TEFA  **1
        TEFB  **1
        LXD  SKFILE-1,4
        CLA*  1,4          PICK UP THE TPE NUMBER
        TSX  $(IOS),4      SET UP THE TAPE ADDRESSES
        LXD  SKFILE-1,4    LOAD IT AGAIN--MAN
        CAL  2,4          IS THERE A SECOND ARGUMENT
        ANA  =077777770000
        ERA  =0007400000000
        TNZ  ONEARG        NO SECOND ARGUMENT
      GOGO CLA*  2,4          PICK UP THE SECOND ARGUMENT
        TZE  BUMP+1        DID SOME DUMMY WANT  NO FILES
      LOOP SUB  =01000000
      RDS  XEC* $(RDS)      READ THE TAPE
        TCDB  *
        TEFA  BUMP        DID WE HIT
        TEFB  BUMP        AN END OF FILE
        TRA  RDS          GO READ SOME MORE
      BUMP TNZ  LOOP
        LXD  SKFILE-3,1
        LXD  SKFILE-2,2
        LXD  SKFILE-1,4
        NOP
        TRCA  **1          TURN OFF TAPE CHECK
        TRCB  **1
        TRA  3,4
      ONEARG CLA  SKFILE-1
        ADD  =01000000      SET UP XR4 FOR PROPER RETURN
        STO  SKFILE-1
        PKD  0,,0          SET UP FOR ONE FILE
        TRA  RDS
      END

COUNT 1200
      REM INPUT ROUTINE USING ARITHMETIC STATEMENTS.  CF NASA TN D-1092
      LBL  INPUT,6
      ENTRY INPUT
      REM THIS IS SUBROUTINE INPUT.  ITS CALLING SEQUENCE
      REM CONTAINS THREE ARGUMENTS---AN IDENTIFICATION
      REM CODE NUMBER, THE FIRST LOCATION RELATIVE TO WHICH
      REM ALL DATA IS TO BE LOADED, AND THE FIRST LOCATION
      REM OF A TABLE TO BE USED BY THE ROUTINE.
      REM
      REM
      REM INCLUDED IN THIS ASSEMBLY ARE SUBROUTINES
      REM 1 INPUT
      REM 2 CHRCTR
      REM 3 CLEAR
      REM 4 COMPAR
      REM 5 ERROR
      REM 6 LOOK
      REM 7 NAME
      REM 8 NUMBR
      REM 9 STORE
      REM 10 TABLE
      REM 11 TEST
      REM 12 ACCUM, FIX, FLT, BINARY
      REM 13 PRINX
      REM 14 READ.
      REM
      INTAPE PZE 0,,7      LEWIS INPUT TAPE NOT STD.
      OUTAPE PZE 0,,6      FORTRAN STANDARD OUTPUT TAPE
      INDX  PZE
        PZE      STORAGE FOR IRA
        PZE      IRB
        PZE      IRC
      INPUT BCI 1,INPUT
        SXD  INUX,1        SAVE INDEX REGISTER A.
        SXD  INUX+1,2      SAVE INDEX REGISTER B.
        SXD  INUX+2,4      SAVE INDEX REGISTER C.
        NZT* 1,4          IF THE IDENTIFICATION NUMBER IS Z
        TRA  4,4          RETURN TO THE CALLING PROGRAM.
        CLA  =1835
        ADD  2,4          2,4 IS THE BASE LOCATION.
        STA  SET
        STA  LOC1
        STA  LOC4
        CLA  TSXBS        OPEN BACKSPACE GATE
        STO* $(LINK)      CALL CHAIN WILL BACKSPACE
        CLA  1,4          1,4 IS THE IDENTIFICATION NUMBER.
      LOCA STA  NREG1
        AXT  36,1        INITIALIZE 36
        STZ  1+1,1        LOCATIONS
        TIX  *-1,1,1      TO ZERO.
        STO  ILOC1        MAKE NON+ZERO.
        CLA  3,4          3,4 IS THE LOCATION OF THE TABLE.
        STA  LOCFC        PREPARE
        STA  NREG1-1
        ADD  =1835        THE
        STA  LOCFA        ARGUMENT STORAGES
        STA  LOCKL
        TSX  CLEAR,4      CLEAR THE VAR REGION.

```

00020
00030
00040
00050
00060
00070
00080
00090
00100
00110
00120
00130
00140
00150
00160
00170
00180
00190
00200
00210
00220
00230
00240
00250
00260
00270
00280

00290
00300
00310
00320
00330
00340
00350
00360
00370
00380
00390
00400
00410
00420
00430
00440
00450
00460
00470
00480
00490
00500
00510
00520
00530
00540
00550
00560
00570

LOCA1	CLA	=0076100000000	INHIBIT READING UNTIL	00580
	STO	READ.	ARRAY RECORD REFRESHED	00590
	AXT	43,2	43 FORCES RECORD TO BE FILLED	00600
	SXD	1,2	IN CHRCTR	00610
	REM	LOOK AT THE FIRST CHARACTER ON THE FIRST CARD		00620
	REM	IN SEARCH OF A \$ SIGN.		00630
LOCAA	TSX	CHRCTR,4		00640
	SUB	=H0000\$0	CHECK FOR A \$ SIGN	00650
LOCA.	STO	WORD		00660
	TSX	COMPAR,4		00670
	OCT	242517630000	D, E, FILE FLAG, T	00680
	LXA	NREG1,4	ZERO IF \$D HAS BEEN READ.	00690
	TXL	**2,4,0		00700
	TXI	**1,2,4	BEFORE \$D ADD 4 TO INDEX 2.	00710
	TXH	ERRU,2,7	JUNK	00720
	TXH	SGNDUT,2,6	\$17 BEFORE \$D. FILE FLAG. OFF	00730
	TXL	**3,2,5	\$E BEFORE \$D	00740
	TSX	READ.+1,4	CRASH READ GATE	00750
	TRA	LOCA1	SHOULD NOW HAVE \$D CARD	00760
	TXH	LOCAD,2,4	FIRST \$D.	00770
	TXH	LOCAJ,2,3	\$T AFTER \$D.	00780
	TXH	LOLCK,2,2	\$17 AFTER \$D. FILE FLAG	00790
	TXH	LOLHG,2,1	\$E AFTER \$D.	00800
	REM			00810
LOCAC	LXA	READ.,4	\$D AFTER \$D. TEST IF BUFFER	00820
	TXL	ERRU,4,0	OVERWRITTEN	00830
	REM	THIS IS THE PROGRAM RETURN.		00840
RTN	LXD	INUX,1	RESET INDEX A.	00850
	LXD	INUX+1,2	RESET INDEX B.	00860
	LXD	INUX+2,4	RESET INDEX C.	00870
	TRA	4,4	RETURN TO CALLING PROGRAM.	00880
	REM	HUNT FOR THE = SIGN OF THE \$ DATA CARD.		00890
LOCAD	CLA	=0076100000000	INHIBIT READING UNTIL	00900
	STO	READ.	\$DATA FIELD SCANNED	00910
	TSX	CHRCTR,4		00920
	TSX	COMPAR,4		00930
	BCI	1,=00000		00940
	TRA	**5,2,2		00950
	TRA	ERRU	JUNK	00960
	TRA	LOCAD	ALPHABETIC	00970
	TRA	ERNR	NUMERIC	00980
	SXD	ALF,4	= SIGN	00990
	REM	USE ALF MODE TO TEST ALL CHARACTERS.		01000
	REM			01010
	REM	COMES HERE WHEN = SIGN HAS BEEN FOUND. GET THE		01020
	REM	IDENTIFICATION NUMBER FROM THE CARD.		01030
LOCAF	LXD	1,4		01040
	TXH	**2,4,43		01050
	TXH	LOLAG,4,42	CARD SCANNED OUT.	01060
	TSX	CHRCTR,4		01070
	TSX	COMPAR,4		01080
	BCI	1, \$+ 0		01090
	TRA	**9,2,2		01100
	TRA	ERRM	JUNK	01110
	TRA	ERRM	ALPHABETIC	01120
LOCAE	TSX	BINARY,4	FORM BIN WD IN VAR	01130
	TRA	LOCAF	BLANK	01140
	SXD	ERSW,2	MINUS SET TO BY PASS.	01150
	TRA	LOCAF	PLUS NO EFFECT.	01160
	STO	SIGN	DOLLARS	01170
	REM	COMES HERE TO CHECK THE REGION CODE AND THE		01180
	REM	VALUE APPEARING ON THE \$DATA CARD.		01190
LOCAG	CLA	VAR	COMMA	01200
	TZE	ERRU	DATA SET NO. MISSING	01210
	ALS	18		01220
	STO	**	SAVE IDENT AT TABLE(1).	01230
	SUB	**	PLACE FIRST ARG IN THIS ADDRESS.	01240
	TNZ	RTN	0 IF CALL CODE = \$DATA CODE	01250
	STZ	ALF	ALF = 0 MEANS NO ALF INFO.	01260
	SXA	NREG1,0	INDICATE \$DATA IS READ.	01270
	REM	INST. BELOW ALSO EXECUTED AT READ., PLACED THERE BY CHRCTR		01280
TSXRD	TSX	READ.+1,4	HERE SNEAK PAST READ. GATE	01290
	SXD	TESTJK,0		01300
	TRA	LOCAN		01310
	REM			01320
	REM	COMES HERE IF IT WAS A \$ TABLE CARD.		01330
LOCAJ	TSX	TABLE,4		01340
	TRA	LOCAN3		01350
	REM			01360
	REM	COMES HERE IF AN ALPHABETIC CHARACTER WAS FOUND.		01370
LOCAK	TSX	NAME,4		01380
	TNZ	SET-1	ZERO MEANS ON LEFT OF = SIGN.	01390
	LXD	JK1,1	IF JK1 DIDNOT INCREASE THEN	01400
TESTJK	TXL	ERRL,1,**	AN = SIGN WAS NOT USED.	01410
	SXD	TESTJK,2	SAVE JK1 FOR NEXT TEST.	01420
	CLA	ILOC	SAVE SIGN OF TABLE ENTRY.	01430
	STO	ILUC1		01440
	TRA	LOCAN2		01450
	REM			01460
	LXD	JK,2	PREPARE TO ACCUMULATE THE NUMBERS	01470
SET	CLA	**2	IN THE PSEUDO ACCUMULATOR.	01480
	STO	TEMP		01490
	CLA	ILUC		01500
	TPL	LOLAM	MINUS MEANS FLOAT THE NUMBER.	01510
	TSX	FLT,4		01520
	TRA	LOLAM		01530
	REM			01540
	REM	COMES HERE IF NUMERIC FIELD.		01550
LOCAL	TSX	NUMBER,4		01560
	STO	TEMP		01570

LOCAM	TSX	ACCUM,4	ACCUMULATE RESULTS IN ACC.	01580
	TSX	CLEAR,4		01590
	LXA	WORD,4		01600
	PXA	0,4	+ WORD IN ACC FOR LOCAR	01610
	TXL	LOCAR,4,58	NOT COMMA	01620
	TXH	LOCAR,4,59	NOT COMMA	01630
	LKD	JK1,2	COMMA	01640
	CLA	ACC		01650
	STZ	ACC	INITIALIZE	01660
	LDQ	ILOC1	IS THIS VARIABLE FIXED POINT.	01670
	TQP	LOC1	NEGATIVE IS FIXED POINT.	01680
	TSX	FIX,4		01690
LOG1	STO	**2	STORE THE NUMBER RELATIVE TO BASE.	01700
LOCAN	LKD	JK1,2		01710
LOCAN1	TXI	*+1,2,1	RAISE STORING INDEX BY ONE.	01720
LOCAN2	SKD	JK1,2	SAVE IT.	01730
LOCAN3	LXD	OPER,1	ANY OPERATORS LEFT OVER.	01740
	TXL	*+3,1,0		01750
ERRL	TSX	ERROR,4		01760
	BCI	1,0(1)		01770
	CLA	ACC	ANY DATA LEFT OVER.	01780
	TNZ	ERRL		01790
	REM			01800
	REM		CALL THIS THE SWITCH HOUSE.	01810
LOGAD	TSX	CLEAR,4		01820
LOGAP	TSX	CHRCR,4		01830
LOGAQ	TSX	COMPAR,4		01840
	BCI	1,,0000		01850
	TRA	*+6,2,2		01860
	TRA	LOCAR	\$D, \$T, OR OPERATORS.	01870
	TRA	LOCAK	ALPHABETIC	01880
	TRA	LOCAL	NUMERIC	01890
	TRA	LOCAT	(SIGN	01900
	TRA	LOCAL	DECIMAL	01910
LOCAR	LKD	OPER,1	ANY OPERATORS LEFT OVER.	01920
	TXH	ERRL,1,0	HIGH MEANS ALREADY HAS OPERATOR.	01930
	SUB	=H0000\$0	SPLIT OFF \$ FROM OTHERS	01940
	TPL	LOCA.	IF + PROCESS \$ TYPE CH	01950
	REM		WHAT KIND OF OPERATOR IS THIS.	01960
	TSX	COMPAR,4		01970
	BCI	1,+/-*,0		01980
	TXH	ERRL,2,5	REMOVE THE JUNK.	01990
	TXH	LOCAN,2,4	COMMA	02000
	SKD	OPER,2	SAVE REST, WILL BRANCH IN SUB ACCU	02010
	TRA	LOLAP	AFTER BOTH OPERANDS HAVE BEEN FOUN	02020
	REM			02030
	REM		COMES HERE IF THE OCT OR ALF MODE.	02040
LOCAT	TSX	CHRCR,4		02050
	TSX	COMPAR,4		02060
	BCI	1,,10A000		02070
	TRA	*+5,2		02080
	TRA	ERRL	JUNK	02090
	TRA	LOCAZ	A CHARACTER	02100
	TRA	LOCAU	D CHARACTER	02110
	REM			02120
	REM		COMES HERE IF EMPTY PARENTHESIS WERE FOUND.	02130
	TSX	CHRCR,4)SIGN, GET NEXT CHARACTER.	02140
	TQP	*+2	MINUS FOR NEW CARD	02150
	TSX	TEST,4	INSERT COMMA IF NEEDED.	02160
	CLA	ILOC1		02170
	STO	ILOC	PREPARE TO GET VALUE OF	02180
	LKD	JK1,2	CURRENT LEFT SIDE.	02190
	TRA	SET		02200
	REM		COMES HERE IF OCTAL MODE.	02210
LOCAU	TSX	CHRCR,4		02220
	SUB	=H000000		02230
	TNZ	LOCAU		02240
	TRA	LOCAW) SIGN	02250
LOCAV	LDQ	VAR		02260
	RQL	3	REPLACE TOP 3 BITS	02270
	LGR	3	BY NEXT OCTAL CHARACTER	02280
	RQL	3	PUT IN BOTTOM OF MQ	02290
	STQ	VAR		02300
	REM		COMES HERE WHEN) IS FOUND.	02310
LOCAM	TSX	CHRCR,4		02320
	TQP	*+2	MINUS FOR NEW CARD	02330
	TSX	TEST,4		02340
	LXA	WORD,4	CHARACTER TO IRC	02350
	TXL	LOCAV,4,7	OCTAL DIGITS	02360
	TXL	ERRJ,4,58	ALPHABETIC, JUNK, 8, 9.	02370
	TXH	ERRJ,4,59	SPLITS {	02380
LOCAK	LKD	JK1,2	COMMA	02390
	CLA	VAR		02400
	TRA	LOC1		02410
	REM		CONVERT THE NUMBER TO BINARY.	02420
LOCAZ	TSX	BINARY,4		02430
	REM			02440
	REM		COMES HERE IF ALF MODE.	02450
LOCAZ	TSX	CHRCR,4		02460
	TSX	COMPAR,4		02470
	BCI	1,,00000		02480
	TRA	*+5,2,2		02490
	TRA	ERRK	JUNK	02500
	TRA	LOCAZ	ALPHABETIC	02510
	TRA	LOCAZ	NUMERIC	02520
	REM		COMES HERE WHEN) IS FOUND	02530

LOCBA	LXA	VAR,1) SIGN	02540
	TNX	ERRK,1,0	ALF COUNT WAS ZERO.	02550
	SXD	ALF,1		02560
	TSX	CLEAR,4		02570
	TSX	CHRCTR,4	PULL THROUGH CHARACTERS AND STORE	02580
	SUB	=017	FILE FLAG, NEVER NEG.	02590
	TZE	ERRB	COUNT WENT PAST E O JOB.	02600
	TSX	STORE,4	THEM ONE AT A TIME.	02610
	TIX	*-4,1,1	GO BACK TILL NCHAR = 1	02620
	LXD	J,1		02630
	LXD	MSHIFT,4		02640
	CAL	BLANK		02650
	LGR	42,4		02660
	ORS	VAR+1,1	FILL IN PARTIAL WORD WITH BLANKS.	02670
LOCBB	AXT	1,4	IRC TO 1	02680
	LXD	JK1,2		02690
	CLA	J	PREPARE TO STORE ALF WDS	02700
	STD	LOCBC1		02710
LOCBC	SXD	JK1,2		02720
	CLA	VAR+1,4		02730
LOC4	STO	**2		02740
	TXI	*+1,4,1	J = J + 1	02750
LOCBC1	TXH	LOLBU,4,**		02760
	TXI	LOCBC,2,1	JK=JK+1	02770
LOCBD	STZ	ALF		02780
	TSX	CLEAR,4		02790
	TSX	CHRCTR,4	LOOK AT NEXT CHARACTER.	02800
	TQP	*+2	MINUS FOR NEW CARD	02810
	TSX	TEST,4	PUT IN COMMA IF NEEDED.	02820
	SUB	=H00000,		02830
	TZE	LOCAN	GO RAISE AND STORE JK1.	02840
	REM			02850
	REM	THESE ARE ERROR CALLS		02860
ERRB	TSX	ERROR,4		02870
	BCI	1,0(B)		02880
ERRD	TSX	ERROR,4		02890
	BCI	1,0(D)		02900
ERRJ	TSX	ERROR,4		02910
	BCI	1,0(J)		02920
ERRK	TSX	ERROR,4		02930
	BCI	1,0(K)		02940
ERRM	TSX	ERROR,4		02950
	BCI	1,0(M)		02960
ERRU	TSX	ERROR,4		02970
	BCI	1,0(U)		02980
	REM			02990
	REM	\$E COMES HERE AFTER \$D		03000
LOCBG	CLA	=0076100000000	NOP	03010
	STO*	\$(LINK)	CLOSE BACKSPACE GATE	03020
	TRA	LOCAC	RETURN	03030
	REM	PURPOSE OF \$END CARD IS TO PROTECT FORIEGN DATA FROM		03040
	REM	BACKSPACE WHEN CHAIN IS CALLED.		03050
	REM			03060
	REM	END OF THE MAIN SEGMENT		03070
	REM	THIS A ROUTINE TO BACKSPACE THE INPUT TAPE WHEN A		03080
	REM	CALL CHAIN IS GOING TO SPILL THE BUFFER.		03090
	REM	THIS ROUTINE IS EXECUTED FROM CHAIN VIA THE ONE		03100
	REM	WORD SUBROUTINE (LINK) WHICH CONTAINS EITHER TSX OR NOP		03110
TSXBS	TSX	LOCBS,4	TO BE STORED AT (LINK)	03120
LOCBS	SXA	*+4,4	SAVE INDEX 4	03130
	CLA	INTAPE	INPUT TAPE NUMBER	
	CALL	\$(IOS)	SELECT INPUT TAPE	03150
	XEC*	\$(BSR)	BACKSPACE IT	03160
	AXT	**4	RESTORE INDEX	03170
	TRA	1,4	RETURN TO THE CHAIN ROUTINE	03180
	EJECT			03190
	REM	THIS IS SUBROUTINE CHRCTR. IT STORES SUCCESSIVE		03200
	REM	CHARACTERS FROM THE CARD AT LOCATION WORD, READS		03210
	REM	SUCCESSIVE CARDS INTO THE ARRAY RECORD, AND PRINTS		03220
	REM	\$\$ TYPE CARDS. THE FIRST CHARACTER FROM A NEW CARD		03230
	REM	IS STORED IN WORD WITH A MINUS SIGN.		03240
	REM			03250
	REM			03260
CHRCTR	SXD	TEMP-10,2		03270
	SXD	TEMP-17,4		03280
	LXD	1,2	CARD COL COUNT, SAW COUNT	03290
	TXH	*+2,2,83	TOO EARLY TO READ.	03300
	XEC	READ.	GATE MAY BE CLOSED	03310
	LQ	Q	HAS UNUSED CHARACTERS FROM BEFORE	03320
	CLA	SIGN	ZERO OR \$ GOES TO TAG	03330
LOCCA	ALS	6	SHIFT LEFT 1 CHARACTER	03340
	SLW	TAG	CLEAR OR PRELOADS TAG	03350
LOCCB	LXD	ALF,4	NONZERO MEANS ALF MODE.	03360
LOCCC	TXH	LOLCD,2,43	SAW COUNT GIVES COL 81 = 43.	03370
	TXH	LOCCG,2,42	WAS COL 80 PROCESSED.	03380
LOCCD	PXD	0,0	CLEAR ACCUMULATOR.	03390
	LGL	6	SHIFT NEXT CHARACTER INTO ACC.	03400
	TIX	LOLCE,2,14	COUNT DOWN BY 14	03410
	LQ	RECORD+3,2	LOAD NEXT WORD	03420
	TXI	*+1,2,69	JUMP BACK COUNTER.	03430
LOCCE	TXH	LOCCF,4,0	RETURN IF ALF MODE.	03440
	PAX	0,1	MOVE CHR. INTO INDEX 1	03450
	TXH	LOLCF,1,48	TRA MEANS GOOD CHARACTER.	03460
	TXH	LOLCC,1,47	TRA IF BLANK	03470
	TXH	LOLCF,1,43	TRA IF GOOD CHARACTER.	03480
	TXL	LOLCF,1,42	TRA IF GOOD CHARACTER.	03490
	ZET	TAG	HERE ON \$	03500
	TRA	PRINT	HERE ON \$\$ GO PRINT	03510
	TRA	LOLCA	\$ GOES TO TAG.	03520
	REM			03530

LUCCF	SXD	I,2	SAVE SAW COUNT	03540
	STQ	Q	SAVE UNUSED CHARACTERS.	03550
	ADD	TAG	ATTACH \$ SIGN IF PRESENT.	03560
	STO	WORD	SAVE THE CHARACTER AT WORD.	03570
	LDQ	SIGN	SIGN OF MQ NEGATIVE IF NEW CARD.	03580
	STZ	SIGN	CLEAR SIGN.	03590
	STZ	TAG	CLEAR TAG OF ANY \$	03600
	LXD	TEMP-17,4		03610
	LXD	TEMP-10,2		03620
	TRA	1,4	RETURN	03630
	REM	PRINT OUT THE \$\$	CARDS.	03640
PRINT	STQ	Q		03650
	XEC*	\$(TES)	CHECK FOR QUIET BUFFERS.	03660
	XEC	READ.	FETCH NEXT CARD.	03670
	LDQ	Q		03680
	LGL	6	SPACE CONTROL SAFE IN ACC	03690
	LDQ	BLANK		03700
	AKT	4,4	FILL END OF OUTPUT	03710
	STQ	OUTBUF+19,4	BUFFER WITH BLANKS.	03720
	TIX	*-1,4,1		03730
	LGR	6	SPACE CONTROL BACK TO MQ.	03740
	STQ	OUTBUF	STORE SPACE CONTROL.	03750
	AXT	14,4		03760
	LDQ	RECORD+2,4		03770
	STQ	OUTBUF+15,4		03780
	TIX	*-2,4,1		03790
	TSX	PRINX,4		03800
	TRA	*+3		03810
LOCCG	XEC	READ.	ALMOST ALWAYS A NOP.	03820
	XEC*	\$(TES)	WAIT FOR QUIET READ BUFFER.	03830
	STZ	TAG	CLEAR THE \$\$ CHARACTERS.	03840
	AXT	14,2	FETCH CARD.	03850
	LDQ	INBUF+14,2	14 WORDS	03860
	STQ	RECORD+2,2		03870
	TIX	*-2,2,1		03880
	CLA	TSXRD	OPEN READ. GATE	03890
	STO	READ.		03900
LOCCJ	AXT	84,2	CARD COL 1 IS 84	03910
	CLS	=0	SET MINUS ZERO IN SIGN	03920
	STO	SIGN		03930
	LGL	12	SAVE COLUMN 79 AND 80	
	LDQ	BLANK	BLANK OUT COLUMN 81 TO 84	
	LGR	12	MAY HAVE LOOK AHEAD	
	STQ	RECORD+1		
	LDQ	RECORD-12		03940
	TRA	LOCCA		03950
	REM			03960
	REM	COMES HERE ON END OF FILE FLAG		03970
LOCCK	LXD	TESTJK,4		03980
	TXH	RTN,4,0	WAS DATA LOADED. YES RTN	03990
SGNOUT	XEC*	\$(TES)	WAIT FOR QUIET OUTPUT BUFFER	04000
	AXT	6,4		04010
	LDQ	OUT+6,4		04020
	STQ	OUTBUF+6,4		04030
	TIX	*-2,4,1		04040
	AXT	13,4		04050
	LDQ	BLANK		04060
	STQ	OUTBUF+19,4		04070
	TIX	*-1,4,1		04080
	TSX	PRINX,4		04090
	XEC*	\$(TES)	WAIT FOR QUIET BUFFER.	04100
LUCOUT	CALL	\$EXIT	THIS WAY OUT FOR KEEPS	04110
	REM			04120
OUT	BCI	6,1	END OF FILE INPUT TAPE JOB COMPLETE	04130
	REM			04150
	REM	END OF THE SAP SUBROUTINE CHRCTR.		04160
	EJECT			04170
	REM	THIS IS SUBROUTINE CLEAR. IT INITIALIZES		04180
	REM	NECESSARY PARAMETERS FOR SUBROUTINE STORE.		04190
	REM			04200
CLEAR	SXD	J,0	SET J TO 0.	04210
	STZ	VAR	CLEAR VAR(1).	04220
	SXD	MSHIFT,0	RESET MSHIFT.	04230
	TRA	1,4	RETURN TO CALLING PROGRAM	04240
	REM			04250
	REM	END OF THE SAP SUBROUTINE CLEAR.		04260
	REM			04270
	REM			04280
	REM	THIS IS FUNCTION COMPAR. IT EXAMINES THE CURRENT		04290
	REM	CHARACTER AND TESTS IT AGAINST THE CHARACTERS		04300
	REM	FOUND IN THE ARGUMENT. ALPHABETIC AND NUMERIC		04310
	REM	SPLITS ARE MADE IF THE CHARACTER IS NOT FOUND		04320
	REM	IN THE ARGUMENT. THESE TESTS ARE COUNTED AND		04330
	REM	THE NUMBER LEFT IN INDEX 2 CORRESPONDS TO THE		04340
	REM	SUCCESSFUL TEST. IF NO TEST IS SUCCESSFUL		04350
	REM	THEN INDEX 2 CORRESPONDS TO THE TOTAL TESTS +1.		04360
	REM			04370
COMPAR	LDQ	1,4	USE FIRST ARGUMENT IN CALLING	04380
	AXT	1,2		04390
LOCCA	PKD	0,0		04400
	LGL	6	PULL IN 1ST TEST CHARACTER.	04410
	TZE	LOCCD	DONE IF ZERO.	04420
	CAS	WORD	CHECK TEST WORD AGAINST CARD	04430
	TXI	LOCCA,2,1	CHARACTER.	04440
	TRA	LOCCD	EQUAL.	04450
LOCCDB	TXI	LOCCA,2,1	NOT EQUAL. GET NEXT TEST	04460
LOCCDC	CLA	WORD	CHARACTER.	04470
	TRA	2,4	PROGRAM RETURN.	04480

LOCDD	CLA	2,4	USE SECOND ARGUMENT IN THE CALLING	04490
	PDX	0,1	SEQUENCE (DECREMENT) AS THE TEST	04500
	TNX	LOCDC,1,1024	FOR ALPHABETIC-NUMERIC SPLIT.	04510
	SXD	LOCDF,1	BECOMES INCREMENT	04520
	LXA	WORD,1	CHARACTER TO IRA	04530
	TXL	LOCDC,1,9	NUMERIC	04540
	TXH	LOCDF,1,57	SPECIAL 0 ZONE, \$X	04550
	TXH	LOCDE,1,49	ALPHABETIC 0 ZONE, NO /	04560
	TIX	**2,1,32	KNOCK OFF 11 ZONE EXCEPT -	04570
	TIX	**1,1,16	KNOCK OFF 12 ZONE EXCEPT +	04580
	REM	+ AND - SIGNS WILL BE (16)10, / WILL BE (17)10		04590
	TXH	LOCDF,1,9	SPECIAL	04600
LOCDE	TXI	LOCDF,2,-1	ADJUST IRB FOR ALPHABETIC	04610
LOCDF	TXI	LOCDC,2,**	ADJUST IRB FOR SPLIT	04620
	REM			04630
	REM	END OF THE SAP SUBROUTINE COMPAR.		04640
	EJECT			04650
	REM	THIS IS SUBROUTINE ERROR. IT IS CALLED IF AN		04660
	REM	ERROR WAS DETECTED ON ANY OF THE INPUT CARDS.		04670
	REM			04680
ERROR	SKA	**2,4	SAVE SOURCE	04690
	XEC*	\$(TES)	WAIT FOR QUIET BUFFERS	04700
	AXT	**4		04710
	CLA	1,4	GET PRINT ARGUMENT	04720
	STO	OUTBUF		04730
	AXT	1,1		04740
	CAS	R		04750
	TRA	**3	S THROUGH V	04760
	TXI	**1,1,-1	R	04770
MES\$A	PXD	BLANK+4,1	A THROUGH N	04780
	ANA	=7817		04790
	ARS	16		04800
	ACL	MES\$A		04810
	STA	102.		04820
	AXT	4,4		04830
102.	LDQ	**4		04840
	STQ	OUTBUF+5,4		04850
	TIX	**2,4,1		04860
	AXT	14,4		04870
	LDQ	RECORD+2,4		04880
	STQ	OUTBUF+19,4		04890
	TIX	**2,4,1		04900
	TSX	PRINX,4		04910
	XEC*	\$(TES)	WAIT FOR QUIET BUFFER	04920
	AXT	19,2		04930
	CLA	BLANK		04940
	STO	OUTBUF+19,2		04950
	TIX	**1,2,1		04960
	LDQ	=H	PICK UP *	04970
	LXD	1,2	SAW COUNT	04980
	TXL	**2,2,71	BACK UP IF OVER 71	04990
	TXI	**3,2,-69		05000
	RQL	6	ROTATE ACCORDING TO CHR PART.	05010
	TIX	**1,2,14	COUNT CHARACTER PART.	05020
	STQ	OUTBUF+19,2	STORE ACCORDING TO RESIDUAL	05030
	TSX	PRINX,4	PRINT THE *	05040
	XEC*	\$(TES)	WAIT FOR THE * TO BE PRINTED	05050
	LXD	ERS,4	PICK UP ERROR SWITCH.	05060
	TXL	LOCOUT,4,0	NON ZERO MEANS TRY NEXT SET	05070
	AXT	1208,4	BYPASS MARK	05080
	SXD	BLANK,4	MARK BYPASSED CARDS	05090
	LXA	NREG1,4	NONZERO IF THIS \$DATA CARD.	05100
	TXL	**2,4,0		05110
	TSX	READ,1,4	CRASH READ GATE IF \$DATA CARD.	05120
LOC\$B	TSX	CHKCTR,4	SKIP TO NEXT \$DATA AND TRY THAT SET.	05130
	TQP	LOC\$B		05140
	SUB	=H0000\$D		05150
	TNZ	LOC\$C	TRA NOT A \$DATA CARD	05160
	STQ*	NREG1-1	PUTS - SIGN IN TABLE(1)	05170
	LXD	BLANK+7,4		05180
	SXD	BLANK,4		05190
	LXD	INDX+2,4		05200
	TRA	LOCA		05210
LOC\$C	ADD	=5	TEST FOR END FILE FLAG	05220
	TZE	SGNOUT	END FILE.. GET OFF	05230
	TRA	LOC\$B	OTHER	05240
	REM			05250
	REM	ERROR MESSAGES. FIRST WORD ALSO USED AS A BLANK.		05260
BLANK	BCI	4,	REDUNDANCY CHECK	05270
	BCI	4,	ILLEGAL CHARACTER	05280
	BCI	4,	NO MANTISSA BEFORE E.	05290
	BCI	4,	NO ENTRY IN TABLE	05300
	BCI	4,	\$TYPE MISSING OR WRONG	05310
	BCI	4,	EXPON. OUT OF RANGE	05320
	REM			05330
	REM	END OF THE SAP SUBROUTINE ERROR.		05340
	EJECT			05350
	REM	THIS IS SUBROUTINE LOOK. IT SEARCHES THE TABLE		05360
	REM	FOR THE NAME STORED AT LOCATION VAR. IF FOUND,		05370
	REM	THE ACC IS NON-ZERO AT THE RETURN.		05380
	REM			05390
LOOK	SXD	TEMP-12,4	SAVE INDEX REGISTER C.	05400
	CLA	J	SUBROUTINE.	05410
	STD	LOC\$E		05420
	AXT	2,2	JK = 2 IN INDEX B	05430
	AXT	1,1	J1 = 1 IN INDEX A	05440
LOC\$A	CAL	**2	CAL TABV(JK).	05450
	TZE	LOC\$G	NO ENTRY THIS VARIABLE	05460
	STD	LOC\$D	DECREMENT HAS NEXT	05470
	ACL	=0377777000000		05480
	ANA	=0377777000000	ENTRY LOC. SAVE DECR	05490
	SUB	J	ONLY. CHECK ENTRY LENGTH.	05500
	TNZ	LOC\$D	IF NOT THE SAME, LOOK AT NEXT ENTR	05510
	PXD	0,2		05520
	PDX	0,4	JM = JK IN INDEX C.	05530

LOCFB	CLA	VAR+1,1	SEE IF VAR AND THIS	05540
LOCFC	CAS	**4	ENTRY AGREE	05550
	TRA	**2	IF SO, CHECK REST OF NAME	05560
	TKI	**2,4,1	RAISE JM BY ONE.	05570
LOCFD	TXI	LOCFA-1,2,**	IF NOT SO, GO TO NEXT ENTRY.	05580
	TXI	**1,1,1	RAISE J1 BY ONE.	05590
LOCFE	TKL	LOCFB,1,**	FINISHED IF J1 IS GREATER THAN J.	05600
	TSX	CLLAR,4	CLEAR IF THE ENTRY AGREES.	05610
LOCFF	CLA*	LOCFA		05620
	STD	ILUC	SAVE COMMON INDEX AT ILOC.	05630
LOCFG	LXD	TEMP-12,4	PREPARE TO RETURN.	05640
	TRA	1,4	RETURN TO THE CALLING PROGRAM.	05650
	REM			05660
	REM			05670
	REM		END OF THE SAP SUBROUTINE LOOK.	05680
	EJECT			05690
	REM		THIS IS SUBROUTINE NAME. IT IS USED TO	05700
	REM		CORRELATE NAMES FROM INPUT CARDS WITH INTERNAL	05710
	REM		MEMORY LOCATIONS BY REFERRING TO THE TABLE.	05720
	REM			05730
	REM			05740
NAME	SXD	TEMP-20,4	SAVE INDEX C.	05750
	REM		GET THE REST OF THE VARIABLE NAME. STOP AT ANY	05760
	REM		NON ALPHANUMERIC CHARACTER.	05770
LOCGB	TSX	STORE,4		05780
LOCGC	TSX	CHRCR,4		05790
	TQP	**2	MINUS FOR NEW CARD	05800
	TSX	TEST,4	COMMA MAY BE NEEDED.	05810
	TNZ	**3	LOOK FOR ZERO. IF ZERO, MAKE IT	05820
	ACL	=H000000	A LETTER O	05830
	STD	WORD		05840
LOCGE	TSX	COMPAR,4		05850
	BCI	1,=10000		05860
	TRA	**5,2,1		05870
	TRA	LOCGF	JUNK OR OPERATORS	05880
	TRA	LOCGB	NUMERIC OR ALPHABETIC	05890
	TRA	LOLGG	(SIGN	05900
	STZ	ILUC1	= SIGN	05910
	REM		GO TO THE TABLE LOOKUP ROUTINE IF AN = SIGN	05920
	REM		OR AN OPERATOR WAS FOUND.	05930
LOCGF	TSX	LOOK,4	FIND THE NAME IN TABLE.	05940
	TZE	ERRT	NAME WAS FOUND IN TABLE IF NON-ZER	05950
	LXA	ILUC,2		05960
	TRA	LOGL		05970
	REM			05980
	REM		GO TO THE TABLE VARIABLE LOOKUP ROUTINE IF A	05990
	REM		(SIGN WAS FOUND.	06000
LOCGG	TSX	LUOK,4		06010
	TNZ	LOLGG		06020
ERRT	TSX	ERROR,4		06030
	BCI	1,0(1)		06040
	REM		CONVERT THE INDEX TO BINARY.	06050
LOCGH	TSX	BINARY,4		06060
	REM		GET THE NUMERICS FOR THE INDEX TO THE VARIABLE.	06070
LOCGJ	TSX	CHRCR,4		06080
	TKL	LULGH,1,9	NUMERIC	06090
	TKL	ERRC,1,27	JUNK	06100
	TXH	ERRC,1,29	JUNK	06110
	TSX	CHRCR,4) SIGN. GET NEXT CHARACTER.	06120
	TQP	**2	MINUS FOR NEW CARD	06130
	TSX	TEST,4	COMMA MAYBE NEEDED.	06140
	TSX	COMPAR,4		06150
	BCI	1,=00000		06160
	TRA	**4,2,1		06170
	TRA	LOLGG	OPERATORS	06180
	TRA	ERRL	ALPHABETIC AND NUMERIC	06190
	STZ	ILUC1	= SIGN	06200
	REM			06210
LOC GK	CLA	VAR	COMPUTE STORING INDEX.	06220
	ACL	ILUC		06230
	PAX	0,2	STORE ADDRESS AT DECREMENT WITHOUT	06240
	TXI	**1,2,-1		06250
LOCGL	SXD	JK,2	ACCUMULATOR OVERFLOW.	06260
	CLA	ILUC1		06270
	LXD	TEMP-20,4	RESTORE INDEX C.	06280
	TRA	1,4	RETURN TO CALLING PROGRAM.	06290
	REM		CONSTANTS AND ERROR CALL.	06300
ERRC	TSX	ERROR,4		06310
	BCI	1,0(1)		06320
	REM			06330
	REM		END OF THE SAP SUBROUTINE NAME.	06340
	EJECT			06350
	REM		THIS IS SUBROUTINE NUMBER. IT IS USED TO	06360
	REM		ASSEMBLE NUMERIC DATA FROM CARDS. ALL VALUES ARE	06370
	REM		TREATED AS FLOATING POINT NUMBERS IN THIS ROUTINE.	06380
	REM			06390
NUMBER	SXD	TEMP-23,4	SAVE INDEX C.	06400
	SXD	KNT2,4	INITIALIZE	06410
	STZ	KNT3	THE SUBROUTINE	06420
	STZ	KNT1	BRANCH PARAMETERS.	06430
	STZ	KNT4		06440
	STZ	TEMP		06450
	TRA	LULNB		06460
	REM			06470
LUCHA	TSX	CHRCR,4		06480
	TQP	**2,	MINUS MEANS FROM NEW CARD	06490
	TSX	TEST,4		06500

LOCHB	TSX	COMPAR,4	06510
	BCI	1,,E0000	06520
	TRA	++6,2,2	06530
	TRA	LOCHK	06540
	TRA	ERRE	JUNK OR AN OPERATOR
	TRA	LOCHC	ALPHABETIC
	TRA	LOLHE	NUMERIC
	CLA	KNT2	E
	TNZ	++3	DECIMAL POINT.
	TSX	ERROR,4	06570
	BCI	1,0(N)	06580
	STZ	KNT2	06590
	STZ	NEXP	ZERO MEANS THIS IS THE SECOND POIN
	TRA	LOCHA	06600
	CLA	NEXP	06610
	ADD	=1035	06620
	STO	NEXP	06630
	LXA	KNT1,1	06640
LOCHD	TXH	LOCHD2,1,10	DO NOT ACCUMULATE PAST 10
LOCHD1	TSX	BINARY,4	06650
	TZE	LOCHA	06660
			CONVERT THE DIGIT TO BINARY.
LOCHD2	TXI	++1,1,1	06670
	SXA	KNT1,1	DO NOT COUNT LEADING ZEROS.
	TRA	LOCHA	06700
			COUNT TOTAL NO. OF DIGITS
	REM	COMES HERE WHEN THE EXPONENT FIELD IS	06710
LOCHE	CLA	KNT1	ENCOUNTERED.
	TNZ	LOCHH	06720
	TSX	ERROR,4	THERE MUST BE AT LEAST ONE DIGIT
	BCI	1,0(S)	BEFORE THE E OF AN E FORMAT NUMBER
	CLA	KNT3	06730
	TRA	++2	06740
			SEE IF EXPONENT DIGITS HAVE ARRIVE
LOCHG	CLS	KNT3	06750
	TNZ	LOCHK-2	06760
	STO	TEMP	SEE IF EXPONENT DIGITS HAVE ARRIVE
	CLA	KNT4	NON ZERO MEANS SIGN IS OPERATOR.
	TNZ	ERRF	06770
	SKD	KNT4,2	STORE SIGN OF EXPONENT.
	TSX	CHRCR,4	06780
LOCHH	TQP	++2,	NONZERO MEANS MORE THAN 1 EXP SIGN
	TSX	TEST,4	MAKE NOZERO.
	TSX	COMPAR,4	06800
	BCI	1,+-,000	06810
	TRA	++7,2,2	06820
	TRA	LOCHK-2	06830
	TRA	ERRF	OTHERS
	TRA	LOCHJ	ALPHABETIC
	TRA	ERRF	NUMERIC
	TRA	LOCHG	DECIMAL
	TRA	LOCHF	MINUS
	REM	CONVERT THE EXPONENT TO BINARY.	06840
LOCHJ	CLA	TEMP	06850
	ALS	2	06860
	ADD	TEMP	06870
	ALS	1	06880
	ACL	WORD	06890
	STO	TEMP	06900
	SXD	KNT3,2	06910
	TRA	LOCHH	06920
	REM	COMES HERE WHEN AN OPERATOR WAS FOUND.	06930
	CLA	KNT3	06940
	TZE	ERRF	TEST FOR THE PRESENCE OF EXPONENT.
LOCHK	CLA	KNT2	06950
	TZE	++2	06960
	STZ	NEXP	06970
	CLA	KNT1	06980
	SUB	=10835	06990
	TPL	++2	SEE IF MORE THAN TEN NUMBERS HAVE
	PXD	0,0	07000
	SUB	NEXP	BEEN CONVERTED
	ADD	TEMP	07010
	STO	NEXP	IF SO, USE THE DIFFERENCE IN THE
	REM	MANTISSA IN VAR AND THE EXPONENT IS IN NEXP.	07020
	CLA	VAR	07030
	TZE	LOCHQ	07040
	LQD	=0233000000000	SHORT CUT IF ZERO.
	LGR	8	07050
	RQL	8	CHARACTERISTIC FOR LOW BITS
	LRS	0	07060
	STQ	VAR	LOW 8 BITS TO MQ
	ORA	=0243000000000	07070
	FAD	VAR	BRING SIGN
	FRN	VAR	07080
	STO	NEXP	07090
	AXT	1,2	CHARACTERISTIC FOR HIGH BITS
	LQD	=1.	07100
	LBT		07110
	TRA	LOCHM	07120
	TXH	ERRV,2,6	07130
	STO	VAR-2	07140
	FMP	TAB+1,2	07150
	XCA	VAR-2	07160
	CLA	VAR-2	07170
	ARS	1	07180
	TZE	LOCHN	07190
	TKI	LOCHL,2,1	07200
LOCHN	TMI	LOCHO	07210
	FMP	VAR	07220
	FRN		07230
	TRA	LOCHQ	07240
	STQ	VAR-2	07250
	CLA	VAR	07260
	FDP	VAR-2	07270
	XCA		07280
	LXD	TEMP-23,4	07290
	TRA	1,4	07300
	REM	THESE ARE THE ERROR CALLS FOR SUB NUMBR.	07310
			07320
			07330
			07340
			07350
			07360
			07370
			07380
			07390
			07400
			07410
			07420
			07430
			07440
			07450
			07460
			07470
			07480
			07490
			07500
			07510
			07520
			07530
			07540
			07550
			07560
			07570

ERRE	TSX	ERROR,4	07580
	BCI	1,0(E)	07590
ERRF	TSX	ERROR,4	07600
	BCI	1,0(F)	07610
ERRV	TSX	ERROR,4	07620
	BCI	1,0(V)	07630
	REM		07640
	REM	THIS IS THE FLOATING PT. TABLE USED IN DBC	07650
	DEC	1E+32,1E+16,1E+8,1E+4,1E+2	07660
TAB	DEC	1E+1	07670
	REM		07680
	REM	END OF THE SAP SUBROUTINE NUMBER.	07690
	EJECT		07700
	REM	THIS IS SUBROUTINE STORE. IT STORES CHARACTERS	07710
	REM	AT THE ARRAY VAR.	07720
	REM		07730
STORE	SXD	TEMP-13,1	07740
	SXD	TEMP-14,2	07750
	LXD	J,1	07760
LOCJA	LXD	MSHIFT,2	07770
	TIX	LOCJB,2,6	07780
	AXT	36,2	07790
	TXI	**1,1,1	07800
	STZ	VAR,1	07810
LOCJB	CLA	WORD	07820
	LDQ	=0	07830
	LGR	42,2	07840
	STQ	TEMP-7	07850
	CAL	TEMP-7	07860
	ORS	VAR+1,1	07870
	SXD	MSHIFT,2	07880
	SXD	J,1	07890
	LXD	TEMP-13,1	07900
	LXD	TEMP-14,2	07910
	TRA	1,4	07920
	REM		07930
	REM	END OF THE SAP SUBROUTINE STORE.	07940
	EJECT		07950
	REM	THIS IS SUBROUTINE TABLE. IT IS USED TO	07960
	REM	CONSTRUCT A TABLE OF NAMES TO BE USED ON CARDS	07970
	REM	AND THEIR MEMORY LOCATIONS RELATIVE TO ARG 2 OF	07980
	REM	THE CALLING SEQUENCE.	07990
	REM		08000
	REM		08010
TABLE	SXD	TEMP-15,4	08020
	STZ	TEMP	08030
LOCKA	TSX	CHRCR,4	08040
	TSX	COMPAR,4	08050
	BCI	1,,00000	08060
	TRA	**5,2,2	08070
	TRA	LOCKD+1	08080
	TRA	LOCKA	08090
	TRA	LOCKD+1	08100
LOCKB	STZ	TEMP	08110
	TRA	LOCKD	08120
	REM	COMES HERE TO CONVERT THE ADDRESS TO OCTAL FOR	08130
LOCKC	CLA	TEMP	08140
	ALS	2	08150
	ADD	TEMP	08160
	ALS	1	08170
	ACL	WORD	08180
	STD	TEMP	08190
	REM		08200
	REM	COMES HERE TO GET NUMERICS.	08210
LOCKD	TSX	CHRCR,4	08220
	TSX	COMPAR,4	08230
	BCI	1,,=000	08240
	TRA	**7,2,2	08250
	TRA	ERRA	08260
	TRA	ERRA	08270
	TRA	LOCKC	08280
	TRA	LOCKT	08290
	TRA	LOCKF	08300
	REM		08310
	REM	COMES HERE IF A DECIMAL PT WAS FOUND.	08320
LOCKE	CAL	TEMP	08330
	CHS		08340
	STD	TEMP	08350
	TRA	LOCKD	08360
	REM		08370
	REM	COMES HERE IF AN = SIGN WAS FOUND.	08380
LOCKF	TSX	CLEAR,4	08390
LOCKG	TSX	CHRCR,4	08400
	TQP	**2,	08410
	TSX	TESTT,4	08420
	TZE	LOCKH	08430
	TSX	COMPAR,4	08440
	BCI	1,,/0000	08450
	TRA	**5,2,1	08460
	TRA	ERRG	08470
	TRA	LOCKJ	08480
	SXD	B,2	08490
	SXD	B,2	08500
	TRA	LOCKK	08510
	REM		08520
	REM	COMES HERE TO STORE CHARACTER.	08530
LOCKH	ACL	=H000000	08540
	STD	WORD	08550
LOCKJ	TSX	STORE,4	08560
	TRA	LOCKG	08570
	REM	COMES HERE AT END OF NAME.	08580

LOCKK	TSX	LOOK,4		08590
	TNZ	LOCKR	GOES TO LOCKR IF THERE IS AN ENTRY	08600
	LXD	J,1		08610
	TXI	**1,1,1	ASSEMBLE KEY	08620
	PXD	0,1	IRB HAS FIRST FREE LOC.	08630
	ACL	TEMP		08640
LOCKKL	SLW	**2	STORE KEY INTO TABLE	08650
	SXD	**1,1	ADVANCE TO END	08660
	TXI	**1,2,**		08670
LOCKKM	CAL	VAR+1,1	MOVE NAME, 0 TO TABLE	08680
LOCKN	XEC	LOCKL	SLW IN TABLE	08690
	TNX	**2,1,1	TRANSFER WHEN DONE	08700
	TXI	LOCKM,2,-1	GO BACK TO FINISH	08710
	ARS	34	KEEP ZONE OF 1ST VAR CH.	08720
	TZE	ERRG	WAS NUMERIC, OR J=0	08730
	REM			08740
	REM	REEXAMINE THE	CUT OFF CHARACTER.	08750
LOCKP	LXD	B,2		08760
	TXH	LOCKB,2,1	COMMA	08770
LOCKQ	LXD	TEMP-15,4	/ CHARACTER	08780
	TRA	1,4	RETURN.	08790
	REM			08800
	REM	COMES HERE TO REPLACE KEY		08810
LOCKR	ANA	=Q37777700000	J+1 IN DECREMENT	08820
	ACL	TEMP	LOCATION AND SIGN	08830
LOCKKS	XEC	LOCKL	SLW IN TABLE	08840
	TRA	LOCKP		08850
LOCKKT	CAL	TEMP	IS / LEGAL	08860
	TZE	LOCKQ	YES	08870
	REM	TRA	ERRA	08880
	REM		NO, NUMERICS WAITING	08890
	REM			08900
ERRA	TSX	ERROR,4		08910
	BCI	1,0(A)		08920
ERRG	TSX	ERROR,4		08930
	BCI	1,0(G)		08940
	REM			08950
	REM	END OF THE SAP SUBROUTINE TABLE		08960
	EJECT			08970
	REM	THIS IS SUBROUTINE TEST. IT LOOKS AHEAD TO CLASSIFY		08980
	REM	A NEW CARD. ACOMMA WILL BE PUT INTO THE CURRENT		08990
	REM	CHARACTER POSITION ONLY IF EITHER (1) THE NEXT		09000
	REM	CARD BEGINS WITH A \$ SIGN FOLLOWED BY SOME OTHER		09010
	REM	CHARACTER OR (2) THE NEXT CARD BEGINS WITH AN		09020
	REM	ALPHABETIC AND AN = SIGN IS FOUND AND IT PRECEEDS		09030
	REM	ALL , \$ AND . CHARACTERS ON THAT CARD.		09040
	REM			09050
	REM			09060
TEST	SXD	TEMP-12,4	SAVE INDEX FOR RETURN.	09070
	SUB	=H0000\$0	TEST FOR A \$ SIGN.	09080
	TPL	LOCLA	POSITIVE MEANS \$ SIGN.	09090
	XEC	READ.	SAFE TO REFILL BUFFER	09100
	TXL	LOCLB,1,16	NUMBERS AND SPECIAL	09110
	TXH	**1,1,33	FIX SD SLASH IS SPECIAL	
	TIX	**1,1,16	MOD OUT ZONE	09130
	TXH	LOCLB,1,9	SPECIALS	09140
	REM		ALPHABETIC COME THRU.	09150
LOCLC	AXT	15,1	SCAN THE CARD.	09160
	LDQ	RECORD+3,1		09170
	TXI	**1,1,69	FOR CHARACTER COUNT	09180
	TXL	LOCLB,1,70	DONE IF WHOLE CARD SCANNED	09190
LOCLD	PXD	0,0	OK TO SEARCH 84 COLUMNS	09200
	LGL	6		09210
	PAX	0,2	ZONE TO IRB	09220
	ANA	=017	KEEP DIGIT	09230
	SUB	=013	DIGIT PART OF ,\$.= CHR.	09240
	TZE	LOCLJ	CHECK ZONE	09250
	TIX	LOCLD,1,14	TRY NEXT CHARACTER	09260
	TRA	LOCLC+1		09270
LOCLJ	AXT	LOCLB,2,15	, \$ NEED NO COMMA	09280
LOCLA	SXD	84,1	84 IS CARD COL 1	09290
	CLA	1,1	RESET CHRCTR TO BEGIN CARD	09300
	STO	Q		09310
	CLA	=H00000,	SUBSTITUTE A COMMA.	09320
	STO	WORD		09330
LOCLB	LXD	TEMP-12,4		09340
	CLA	WORD	IN AC FOR SR NAME, TABLE	09350
	TRA	1,4	RETURN TO THE CALLING PROGRAM.	09360
	REM			09370
	REM			09380
	REM	FOR TABLE SUB STATEMENTS		
TESTT	SXD	TEMP-12,4		09390
	AXT	84,4	IF NEXT CARD HAS VALID	09400
	LDQ	RECORD-12	LEFT PART OF SUBSTATEMENT	09410
LOCNB	PKD	0,0		09420
	LGL	6		09430
	PAX	0,2		09440
	TXH	LOCLB,2,48	0 ZONES EXCEPT BLANK	09450
	TXH	LOCNC,2,47	BLANK	09460
	TXH	LOCLB,2,27	11 ZONES AND 1	09470
	TXH	LOCLA,2,26	.	09480
	TXH	LOCLB,2,11	12 ZONES AND 8-4	09490
	TXH	LOCLA,2,10	=	09500
LOCNC	TIX	LOCNB,4,14	NUMERICS AND BLANK	09510
	LDQ	RECORD+3,4		09520
	TNX	LOCLB,4,1		09530
	TXI	LOCNB,4,70		09540
	REM			09550
	REM	END OF THE SAP SUBROUTINE TEST.		09560
	EJECT			09570
	REM	THE FOLLOWING FOUR SUBROUTINES ARE USED TO		09580
	REM	CONVERT DECIMAL DIGITS TO BINARY IN VAR,		09590
	REM	FIX FLOATING POINT NUMBERS, FLOAT FIXED POINT		09600
	REM	NUMBERS, AND FORM ARITHMETIC RESULTS IN THE		09610
	REM	PSEUDO ACCUMULATOR (ACC) FOR EACH OPERATION		09620
	REM	ON A CARD.		09630
	REM			09640

BINARY	CLA	VAK	ACCUMULATE A SERIES OF BASE 10	09650
	ALS	2	DIGITS IN BINARY IN VAR.	09660
	ADD	VAR		09670
	ALS	1		09680
	ACL	WORD		09690
	STO	VAR		09700
	TRA	1,4		09710
	REM			09720
FLT	CLA	TEMP	CONVERT TO FLOATING POINT THE	09730
	LRS	18	CONTENTS OF THE STORAGE CALLED	09740
	ORA	=02330000000000	TEMP.	09750
	FAD	=02330000000000		09760
	STO	TEMP	LEAVE THE ANSWER IN TEMP.	09770
	TRA	1,4		09780
	REM			09790
FIX	UFA	=02330000000000	CONV TO FIXED PT THE CONT	09800
	LRS	0	OF THE ACCUMULATOR.	09810
	ANA	=0377777		09820
	LLS	0		09830
	ALS	18	LEAVE THE FIXED POINT NUMBER IN	09840
	TRA	1,4	THE ACCUMULATOR.	09850
	REM			09860
ACCUM	LXD	OPER,2	BRANCH FOR OPERATOR	09870
	STZ	OPER	PREPARE FOR NEXT OPERATOR.	09880
	CLA	TEMP		09890
	TRA	=+5,2		09900
	TRA	LOCMB	*	09910
	TRA	LOCMA	/	09920
	CHS		MINUS	09930
	FAD	ACC	PLUS	09940
ACCUM	STO	ACC	NONE	09950
	TRA	1,4		09960
	REM			09970
LOCMA	CLA	ACC	DIVIDE.	09980
	FDP	TEMP		09990
	STO	ACC		10000
	TRA	1,4		10010
	REM			10020
LOCMB	LDQ	ACC	MULTIPLY.	10030
	FNP	TEMP		10040
	TRA	ACCUM		10050
	REM			10060
	REM			10070
	REM		END OF THE SAP SUBROUTINES ACCUM, FIX, FLOAT.	10080
	EJECT			10090
	REM	SUBROUTINE PRINX	DRAINS PRINT BUFFER TO LOGICAL TAPE	
	REM		GIVEN IN DECREMENT OF OUTAPE	
PRINX	SKA	PRINY,4	BUFFERED WRITE ROUTINE	10110
	CLA	OUTAPE	LOGICAL OUTPUT TAPE NUMBER	
	CALL	(IOS)		10130
	AXC	IOCD,4		10140
	XEC*	\$(WRS)	=WTDL 6	10150
	XEC*	\$(RCH)	=RCHL 0,4	10160
	PKA	0,4	SAVE LOC OF	10170
	STA*	\$(WTC)	IO COMMAND FOR (WER)	10180
	CLA	TSXMR	PRESET END ACTION	10190
	STO*	\$(TES)		10200
PRINY	AXT	=+4		10210
	TRA	1,4		10220
TSXMR	CALL	(WER)	EXECUTED FROM (TES)	10230
IOCD	IORT	OUTBUF,19		10240
	EJECT			10250
	REM	SUBROUTINE READ.	FILLS READ BUFFER FROM LOGICAL TAPE 7.	10260
READ.	TSX	READ,1,4	READ. GATE INITIALLY OPEN	10270
	CLA	=00761000000000	CLOSE READ GATE	10280
	STO	=-2		10290
	SKA	AXT,4		10300
	CLA	INTAPE	LOGICAL INPUT TAPE NO.	
	CALL	(IUS)		10320
	AXC	IOCD,4		10330
	XEC*	\$(RDS)	=RTDLS	10340
	XEC*	\$(RCH)	=RCHL 0,4	10350
	CLA	TSXTS	SET UP BUFFER TEST	10360
	STO*	\$(TES)		10370
AXT	AXT	=+4		10380
	TRA	1,4		10390
TSXTS	TSX	=+1,4	EXECUTED FROM (TES)	10400
	SKA	AXT,4	CLOSE OUT BUFFER	10410
	CLA	=00761000000000	SAY BUFFER IS QUIET	10420
	STO*	\$(TES)		10430
	AXT	5,4	PRESET REDUNDANCY	10440
	SKA	RTT,4	COUNT	10450
TSKTT	AXC	=+1,4		10460
	XEC*	\$(TCO)	=TCOL 0,4	10470
	AKC	RTT,4		10480
	XEC*	\$(TRC)	=TRCL 0,4	10490
	AKC	XIT,4		10500
	XEC*	\$(TEF)	=TEFL 0,4 JOB COMPETE	10510
	TRA	AXT	RETURN	10520
RTT	AXT	=+4	INTERROGATE COUNT	10530
	TIX	SKA,4,1	GIVE ANOTHER TRY	10540
	AXT	14,4	CARD SURE BAD	10550
	CLA	INBUF+14,4	SAVE IMAGE	10560
	STO	RECORD+2,4		10570
	TIX	=-2,4,1		10580
	AXT	84,4	MAKE ERROR ROUTINE LOSE*	10590
	SKD	1,4	IN INPUT BUFFER	10600
	TSX	ERROR,4		10610
R	BCI	1,0(R)		10620
SKA.	SKA	RTT,4	SAVE COUNT	10630
	AXC	=+2,4		10640
	XEC*	\$(TEF)	TURN OFF EOF IND	10650
	XEC*	\$(BSR)	BACKSPACE,	10660
	AKC	IOCD,4		10670
	XEC*	\$(RDS)	REREAD	10680
	XEC*	\$(RCH)		10690
	TRA	TSXTT		10700

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546